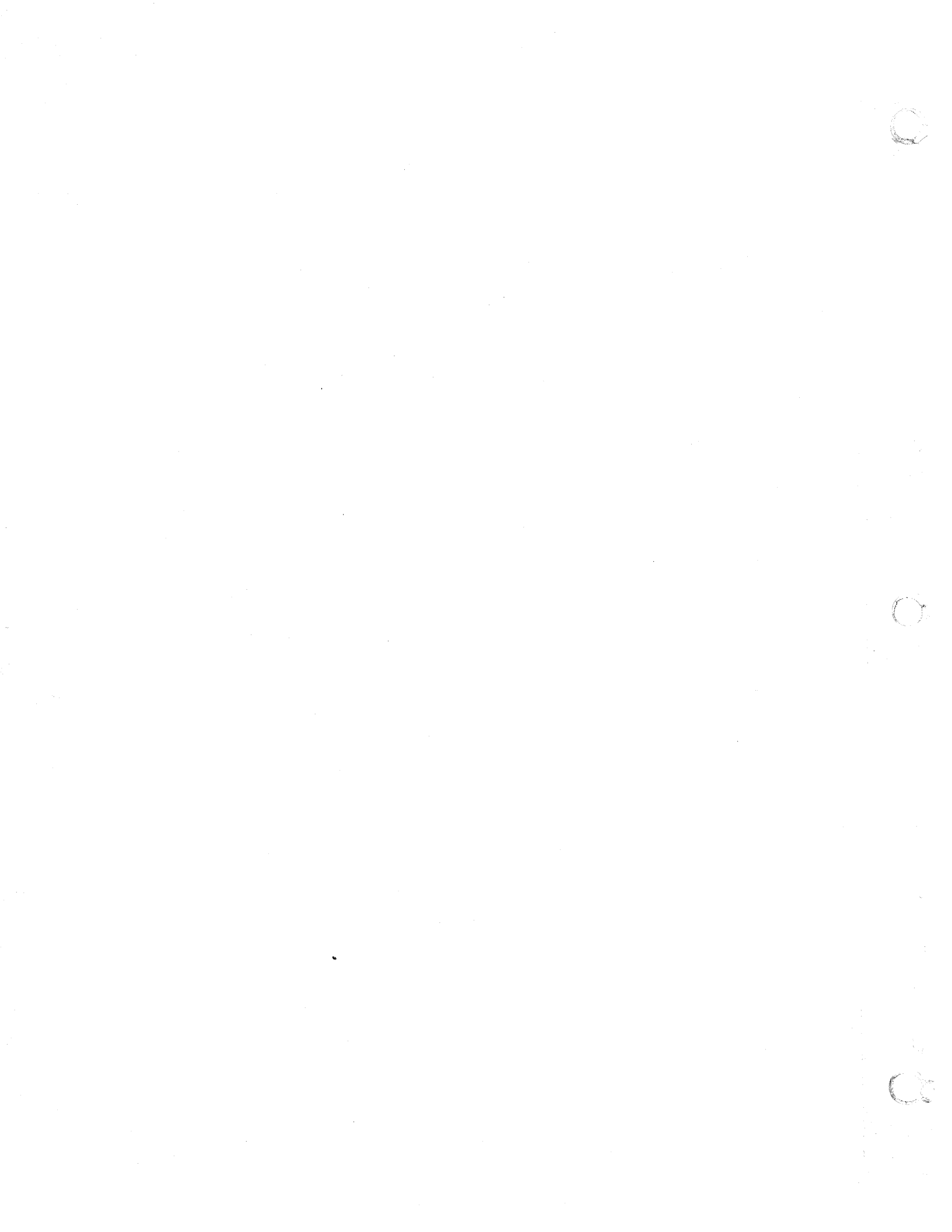


IDENTIFICATION

\*\*\*\*\*

PRODUCT CODE: MAINDEC-08-DHTMA-A-D  
PRODUCT NAME: TMB-E CONTROL TEST PART 1  
DATE CREATED: DECEMBER 4, 1972  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: LEONARD E. BEYERSDORFER

COPYRIGHT (C) 1972  
DIGITAL EQUIPMENT CORPORATION  
MAYNARD, MASS. 01754



NOTE  
-----

THERE ARE SIX DIAGNOSTIC PROGRAMS ASSOCIATED WITH THE TM8-E DECMAGTAPE CONTROL AND ITS TRANSPORT SYSTEM. ALTHOUGH PHYSICALLY SEPARATE, THESE PROGRAMS MUST BE TREATED AS A LARGE INTEGRATED TEST, AND TO ENSURE PROPER SYSTEM OPERATION, THESE TESTS MUST BE EXECUTED IN THE ORDER DELINEATED BELOW.

IF A GIVEN TEST SHOULD FAIL AND IT APPEARS THAT A FIX HAS BEEN FOUND, ALL PROGRAMS MUST ONCE AGAIN BE RUN, ONLY WHEN ALL TESTS HAVE RUN WITHOUT ANY UNACCEPTABLE ERRORS CAN THE TM8-E SYSTEM BE CONSIDERED UP.

TM8-E DIAGNOSTIC PROGRAMS' ORDER OF EXECUTION  
-----

1. TM8-E CONTROL TEST PART 1 (MAINDEC-08-DHTMA)
2. TM8-E CONTROL TEST PART 2 (MAINDEC-08-DHTMB)
3. TM8-E DRIVE FUNCTION TIMER (MAINDEC-08-DHTMC)
4. TM8-E DATA RELIABILITY 9 TRACK (MAINDEC-08-DHTMD)
5. TM8-E DATA RELIABILITY 7 TRACK (MAINDEC-08-DHTME)
6. TM8-E RANDOM EXERCISER (MAINDEC-08-DHTMF)

TABLE OF CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	HARDWARE
2.2	MEMORY
2.3	PRELIMINARY PROGRAMS
3.	CONVENTIONS
4.	PROGRAM LOADING PROCEDURE
5.	PROGRAM STARTING PROCEDURE
6.	STANDARD TEST PROCEDURE
7.	PROGRAM CONTROLS
7.1	PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS
8.1.1	TEST SELECTION COMMANDS
7.1.2	TEST CONTINUATION COMMANDS
7.1.3	TEST INTERRUPT COMMAND
7.1.4	MISCELLANEOUS COMMANDS
7.2	SWITCH REGISTER CONTROL OPTIONS
8.	ERRORS
8.1	ERROR HALTS
8.2	ERROR REPORTS
8.3	STANDARD ERROR RECOVERY PROCEDURE
8.3.1	SUBTEST LOOPS
9.	RESTRICTIONS
10.	EXECUTION TIME
11.	TEST ABSTRACTS
12.	LISTING



1. ABSTRACT  
-----

THE TM8-E CONTROL TEST PART 1 IS AN INTEGRATED SERIES OF SUB-TESTS DESIGNED TO AID IN THE CHECKOUT AND MAINTENANCE OF THE TM8-E DECMAGTAPE CONTROL WITHOUT DEPENDENCE ON TRANSPORT SYSTEM USE. VERSATILITY OF USAGE IS AFFORDED THROUGH A MODEST KEYBOARD MONITOR AND SWITCH REGISTER CONTROL OPTIONS.

THIS PROGRAM CONSISTS OF 12 MAJOR TESTS (TEST 01 THROUGH TEST 14) EACH OF WHICH CONSISTS OF A NUMBER OF SUBTESTS DESIGNATED BY THE LETTERS A THROUGH Z.

2. REQUIREMENTS  
-----

2.1 HARDWARE  
-----

PDP-8/E, 8/M, 8/F  
TELETYPE OR COMPATIBLE DEVICE (TTY)  
TM8-E DECMAGTAPE CONTROL

2.2 MEMORY  
-----

THIS PROGRAM REQUIRES 4K OF MEMORY AND MAY RESIDE IN ANY MEMORY FIELD; ALL OF EXISTING MEMORY IS USED TO TEST THE ABILITY OF THE TM8-E TO ADDRESS CORRECTLY DURING DMA TRANSFERS.

2.3 PRELIMINARY PROGRAMS  
-----

ALL PROCESSOR/MEMORY DIAGNOSTICS

3, CONVENTIONS  
-----

IN THE DESCRIPTION OF ANY KEYBOARD COMMANDS GIVEN IN THIS MANUAL, THE BACK ARROW (←) CORRESPONDS TO DEPRESSING THE RETURN KEY, AND THE NUMBER SIGN (#) CORRESPONDS TO DEPRESSING THE LINE-FEED KEY.

THE PROGRAM PRINTS A LEFT BRACKET (L) WHEN IT IS READY TO ACCEPT A KEYBOARD COMMAND.

4, PROGRAM LOADING PROCEDURE  
-----

LOAD THE PROGRAM INTO ANY DESIRED MEMORY FIELD USING THE STANDARD BINARY LOADER TECHNIQUE.

5, PROGRAM STARTING PROCEDURE  
-----

- A. LOAD ADDRESS 0200.
- B. LOAD THE EXTENDED ADDRESS WITH THE PROGRAM FIELD.
- C. SET THE SR TO 4000.
- D. DEPRESS CLEAR, THEN CONTINUE.
- E. THE PROGRAM WILL PRINT ITS TITLE AND MAINDEC NUMBER, THEN ASK "EXTENDED MEMORY?(0-7)". TYPE THE NUMBER OF THE HIGHEST EXISTING MEMORY FIELD, THEN THE PROGRAM MONITOR WILL ASSUME CONTROL. GO TO THE STANDARD TEST PROCEDURE IN PARAGRAPH 6.

NOTE: THE PROGRAM MAY BE RESTARTED AT ANY TIME AT ADDRESS 0201. IN THIS CASE THE PROGRAM BYPASSES ALL PROGRAM INITIALIZATION AND GOES DIRECTLY TO THE PROGRAM MONITOR TO ACCEPT KEYBOARD COMMANDS.

6. STANDARD TEST PROCEDURE  
-----

USE OF THE STANDARD TEST PROCEDURE ENSURES PROPER TM8-E CHECKOUT. ANY ERROR OCCURRENCE RESULTS IN AN ERROR REPORT ON THE TTY AND IN A RETURN TO THE PROGRAM MONITOR. ALL OPERATIONAL TEST PROCEDURES AFFORDED BY PROGRAM MONITOR AND SWITCH REGISTER CONTROL ARE DESCRIBED IN PARAGRAPH 7. ERROR RECOVERY PROCEDURES AND RELATED INFORMATION ARE GIVEN IN PARAGRAPH 8.

ACCOMPLISH THE FOLLOWING STEPS.

- A. START THE PROGRAM AS DESCRIBED IN PARAGRAPH 5.
- B. IF A TU10 TRANSPORT SYSTEM IS CONNECTED, ENSURE THAT ALL DRIVES ARE OFF LINE. ALL LEVELS INPUT FROM THE TU10 TO THE TM8-E SHOULD BE AT A "HIGH" LEVEL.
- C. SET THE SR=0001.
- D. TYPE "TA" WHICH RESULTS IN THE EXECUTION OF TEST 01 THROUGH TEST 14. SEVERAL PASSES WILL BE MADE OF EACH TEST.
- E. AFTER ALL TESTS HAVE BEEN EXECUTED, THE PROGRAM WILL PRINT "PASS 0001". THEN ANOTHER PASS IS AUTOMATICALLY STARTED.
- F. ALLOW THE PROGRAM TO RUN FOR AT LEAST 2 PASSES. EACH PASS TAKES APPROXIMATELY 15 MINUTES.

7. PROGRAM CONTROLS

THE FOLLOWING SUBPARAGRAPHS DESCRIBE THOSE CONTROLS WHICH THE USER HAS OVER THE PROGRAM.

THERE ARE TWO MAIN SOURCES OF PROGRAM CONTROL: A) PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS; AND B) SWITCH REGISTER CONTROL.

7.1 PROGRAM MONITOR CONTROL VIA KEYBOARD COMMANDS

THE PROGRAM MONITOR KEYBOARD COMMANDS ARE DESCRIBED BELOW. IF A COMMAND ERROR IS DETECTED, MONITOR PRINTS "P" AND THE COMMAND MUST BE RETYPED. COMMANDS MAY BE INPUT AFTER MONITOR HAS PRINTED A LEFT BRACKET ([). THE TEST INTERRUPT COMMAND MAY BE TYPED AT ANY TIME.

THE PROGRAM MONITOR IS ENTERED UNDER ANY OF THE FOLLOWING CONDITIONS.

- A. AFTER PROGRAM STARTUP WITH SR 0=1, OR A RESTART AT 0201.
- B. AN ERROR OCCURS AND SR2=0.
- C. ALL SELECTED TESTS ARE DONE.
- D. THE TEST INTERRUPT COMMAND (REFERENCE PARAGRAPH 7.1.3) IS TYPED BY THE USER.

7.1.1 TEST SELECTION COMMANDS

THE FOLLOWING COMMANDS ARE USED TO SELECT FROM ONE TO TWELVE TESTS FOR EXECUTION, REGARDLESS OF THE ORDER IN WHICH A TEST SELECTION IS MADE; THOSE TESTS ARE EXECUTED IN NUMERICAL ORDER, IF A GIVEN TEST IS SPECIFIED TWICE IN THE SAME SELECTION, IT WILL BE DELETED FROM THAT SELECTION.

COMMAND	RESULT
TA	RUN ALL TESTS (TEST 01 THROUGH TEST 14)
T01110TNN	RUN THE TEST(S) INDICATED, T01 THROUGH T14 IN OCTAL ARE THE VALID SELECTIONS. IF THE USER SELECTS OTHER THAN T01 THROUGH T14 THE PROGRAM WILL EXECUTE ONE OF THE VALID TESTS (UNDETERMINED).

7.1.2 TEST CONTINUATION COMMANDS  
-----

THE FOLLOWING COMMANDS ARE USED TO CONTINUE IN THE TEST SEQUENCE IF THAT SEQUENCE HAS BEEN INTERRUPTED BY AN ERROR WITH SR2=0.

COMMAND	RESULT
-----	-----
C+	CONTINUE IN THE TEST SEQUENCE. IF NO TESTS ARE SELECTED, MONITOR WILL PRINT "?". IF THIS OCCURS A NEW TEST SELECTION MUST BE MADE.
EX+	CONTINUE WITH THE NEXT SUBTEST IF THE CURRENT SUBTEST IS FAILING AND SEVERAL SETS OF DATA REMAIN TO BE USED IN THAT SUBTEST. THIS COMMAND ENABLES THE USER TO EXIT A FAILING DATA SUBTEST AND CONTINUE WITH THE NEXT SUBTEST. AFTER THE CURRENT "EX+" COMMAND HAS BEEN UTILIZED FOR A FAILING DATA SUBTEST EXIT, IT IS NO LONGER EFFECTIVE. IF NO TESTS ARE SELECTED, MONITOR WILL PRINT "?". IN THIS CASE A NEW TEST SELECTION MUST BE MADE.

7.1.3 TEST INTERRUPT COMMAND

THE FOLLOWING COMMAND MAY BE USED TO INTERRUPT TESTING AND RETURN TO THE PROGRAM MONITOR.

COMMAND RESULT

ALTMODE KEY INTERRUPT TEST EXECUTION, RESPONDS TO "ALT" AND "ESC" KEYS. TEST SELECTION IS UNAFFECTED.

7.1.4 MISCELLANEOUS COMMANDS

COMMAND RESULT

TR TRACE THE PROGRAM FLOW USING THE ERROR REPORT FORMAT (MODIFIED). THIS COMMAND IS A PREFIX TO ALL "TEST SELECTION" AND "TEST CONTINUATION" COMMANDS, AND RESULTS IN A MODIFIED ERROR REPORT FOR EACH SUBTEST THAT PASSES. TRACE REPORTS ARE CONTROLLED VIA SR BITS 3 AND 4 IN THE SAME MANNER AS ERROR REPORTS. "TR" REMAINS IN EFFECT UNTIL THE NEXT ENTRY TO THE PROGRAM MONITOR.

DMINNN

DUMP THE CONTENTS OF MEMORY LOCATION NNNN IN FIELD M ON THE TTY. LINE-FEED DUMPS THE CONTENTS OF THE NEXT LOCATION; CARRIAGE RETURN RETURNS TO THE PROGRAM MONITOR, TEST SELECTION IS UNAFFECTED.

7.2 SWITCH REGISTER CONTROL OPTIONS

SR BIT	STATE	FUNCTION
0	0	RUN ALL TESTS NORMALLY.
	1	QUICK VERIFY (ONE PASS IS MADE OF EACH SELECTED TEST).
1	0	ENABLE TEST IN PROGRESS REPORTS, AS SOON AS A TEST IS STARTED THE TEST NUMBER (TESTNN) IS PRINTED ON THE TTY.
	1	DISABLE TEST IN PROGRESS REPORTS.
2	0	RETURN TO PROGRAM MONITOR ON ANY ERROR OCCURRENCE.
	1	RETURN TO PROGRAM MONITOR ONLY IF A FATAL ERROR HAS OCCURRED.
3	0	ENABLE ERROR AND TRACE REPORTS.
	1	DISABLE
4	0	ERROR AND TRACE REPORTS INCLUDE APPLICABLE DATA ONLY.
	1	FORCE ERROR AND TRACE REPORTS TO INCLUDE ALL POSSIBLE DATA.
5	0	DISABLE SUBTEST LOOP.
	1	LOOP ON CURRENT SUBTEST. ALL VARIABLE DATA REMAINS CONSTANT.
6-9	0	INHIBIT LOOPS 6-9 RESPECTIVELY.
	1	ENABLE LOOPS 6-9 RESPECTIVELY. LOOPS 6-9 ALLOW THE USER TO LOOP ON A SMALL GROUP OF SUBTESTS WITHIN THE CURRENT TEST THEREBY ENABLING INCREASING THE TESTING FREQUENCY OF A SMALL SET OF FUNCTIONS. THE LISTING MUST BE REFERENCED TO DETERMINE WHICH SUBTESTS ARE INCLUDED IN EACH LOOP. THE END OF EACH LOOP IS MARKED AS "*****LOOP N*****".
10	0	INHIBIT LOOP ON CURRENT TEST (TEST 01 = TEST 14).
	1	ENABLE LOOP ON CURRENT TEST.
11	0	TERMINATE TESTING UPON COMPLETION OF ALL SELECTED TESTS AND DELETE TEST SELECTION.
	1	RUN ALL SELECTED TESTS CONTINUOUSLY. AT THE COMPLETION OF EACH PASS THROUGH THE ENTIRE TEST SELECTION, THE NUMBER OF PASSES EXECUTED (IN OCTAL) IS PRINTED (PASS NNNN).

8. ERRORS

8.1 ERROR HALTS

THERE ARE NO ERROR HALTS IN THIS PROGRAM, IF AN ERROR SHOULD OCCUR AND SR2=0, THE PROGRAM WILL STOP TESTING AND RETURN TO THE PROGRAM MONITOR TO AWAIT A USER COMMAND. REFER TO PARAGRAPH 7.1 FOR AVAILABLE COMMANDS.



6.2 ERROR REPORTS

WHEN SR4=0, ERROR REPORTS INCLUDE ONLY THAT INFORMATION WHICH APPLIES DIRECTLY TO THE ERROR. IF SR4=1, ALL POSSIBLE INFORMATION IS PRINTED WHETHER APPLICABLE OR NOT. AN EXAMPLE OF A MAXIMUM INFORMATION ERROR REPORT IS SHOWN BELOW.

\*ER01D PC:0250 I1:6716 GD:0000 BD:7777 OD:7777  
WC:1234 CA:2345 CM:0000 FS:0000 MS:0000 DB:0000 AC:0000

THE SYMBOLS USED IN THE ERROR REPORTS ARE DEFINED BELOW.

SYMBOL DEFINITION

\*ERNXX ERROR OCCURRED IN TEST NN, SUBTEST X, (IF NN=00, THE ERROR OCCURRED OUTSIDE OF A FORMAL TEST.)

\*FENNX FATAL ERROR

TR\*NX TRACE REPORT INDICATOR, NO ERROR OCCURRED BUT TRACE IS ENABLED.

PC\*NNNN ADDRESS IN PROGRAM AT WHICH ERROR WAS DETECTED.

I1\*NNNN OCTAL CODE FOR IOT1 IN A VARIABLE SUBTEST.

I2\*NNNN OCTAL CODE FOR IOT2 IN A VARIABLE SUBTEST.

THE FOLLOWING THREE SYMBOLS ARE FURTHER DEFINED IN THE PROGRAM LISTING IN THE SUBTEST COMMENTS.

GD\*NNNN GOOD TEST VALUE

BD\*NNNN REAL TEST VALUE (BAD)

OD\*NNNN PREVIOUS GOOD TEST VALUE (OLD)

THE FOLLOWING SYMBOLS INDICATE THE CONTENTS OF THE SPECIFIED REGISTER AT THE TIME THE ERROR WAS DETECTED.

WC\*NNNN WORD COUNT REGISTER

CA\*NNNN CURRENT ADDRESS REGISTER

CM\*NNNN COMMAND REGISTER

FS\*NNNN FUNCTION/STATUS REGISTER

MS\*NNNN MAIN STATUS REGISTER

DB\*NNNN DATA BUFFER REGISTER

AC\*NNNN PROCESSOR ACCUMULATOR (VALID ONLY FOR ILLEGAL SKIP ERRORS.)

8,3 STANDARD ERROR RECOVERY PROCEDURE  
-----

THE STANDARD ERROR RECOVERY PROCEDURE ASSUMES THAT THE STANDARD TEST PROCEDURE IS BEING USED; THAT IS, THAT ALL ERROR OCCURRENCES RESULT IN AN ERROR REPORT AND IN A RETURN TO THE PROGRAM MONITOR.

WHEN AN ERROR OCCURS, USE THE FOLLOWING STEPS AS A GUIDE FOR RECOVERY REFERRING TO PARAGRAPH 8,2 FOR ERROR REPORT SYMBOL DEFINITIONS.

- A. REFERENCE THE POINT IN THE PROGRAM LISTING INDICATED BY THE "PC:" NUMBER.
- B. THE ERROR CODE IN THE LISTING SHOULD MATCH THE CODE IN THE ERROR REPORT (ERNX OR FENNX).
- C. COMMENTED IMMEDIATELY BELOW THE ERROR CODE IN THE LISTING IS AN EXPLANATION OF THE NUMBERS PRINTED AFTER THE GD, BD AND DD SYMBOLS.
- D. IN THE IMMEDIATE VICINITY OF THE ERROR CODE AND RELATED SUBTEST A DESCRIPTION OF THE SUBTEST WILL BE FOUND.
- E. IF THE ERROR IS FATAL (FENNX), THERE IS NO STANDARD RECOVERY. THE CAUSE OF THE FAILURE MUST BE DETERMINED THROUGH STATIC MEANS.
- F. IF THE ERROR IS NON-FATAL (ERNX), THE USER MAY ELECT TO CONTINUE IN THE TEST SEQUENCE (C+), OR TO ENTER A SUBTEST LOOP AS DESCRIBED BELOW.

SUBTEST LOOPS  
-----

TO ENTER A SUBTEST LOOP, CARRY OUT THE FOLLOWING STEPS:

- A. SET SR BITS 2, 3 AND 5#1.
- B. TYPE "C+".

TO EXIT A SUBTEST LOOP, RESTORE THE SWITCHES TO THEIR NORMAL POSITION.

9. RESTRICTIONS  
-----

NONE.

10. EXECUTION TIME  
-----

ONE NORMAL PASS OF ALL TESTS TAKES APPROXIMATELY 15 MINUTES.  
A QUICK VERIFY PASS (SRQ=1) TAKES ONLY 2 MINUTES.

11. TEST ABSTRACTS  
-----

TEST 01 VERIFIES THE EXISTENCE OF THE VARIOUS 670X AND 671X IOT'S EXCEPT RMSR (6714), AND TESTS THE BASIC OPERATION OF ALL TMS-E REGISTERS EXCEPT THE MAIN STATUS AND STATUS PORTION OF THE FUNCTION/STATUS REGISTERS.

TEST 02 IS A WORST CASE DATA TEST. DATA IS LOADED INTO AND READ BACK FROM ALL NON-STATUS REGISTER BITS AND VERIFIED.

TEST 03 VERIFIES THE EXISTENCE OF THE VARIOUS FUNCTIONS (AS FAR AS POSSIBLE WITHOUT A TRANSPORT) BY CHECKING EACH FUNCTION'S AFFECT ON CURRENT ADDRESS AND WORD COUNT INCREMENTING, AS WELL AS VERIFYING THAT DATA IS TRANSFERRED BETWEEN THE CONTROL AND THE PROCESSOR IN THE PROPER DIRECTION DURING DMA TRANSFERS.

TEST 04 TESTS THE ABILITY OF WORD COUNT, CURRENT ADDRESS AND COMMAND REGISTER BITS 6-8 TO INCREMENT PROPERLY.

TEST 05 VERIFIES THAT PROPER DATA TRANSFERS ARE MADE BETWEEN THE DATA BUFFER AND MEMORY IN BOTH DIRECTIONS.

TEST 06 VERIFIES THAT THE CURRENT ADDRESS REFERENCES THE CORRECT MEMORY LOCATION AND THAT COMMAND REGISTER BITS 6-8 REFERENCE THE CORRECT MEMORY FIELD. ALL EXISTING FIELDS ARE UTILIZED IN THIS TEST.

TEST 07 TESTS THE VARIOUS STATUS BITS TO CONTAIN THE PROPER STATUS AS GOVERNED BY THE "NO TRANSPORT" CONDITION.

TEST 10 CHECKS VARIOUS POSITIVE AND NEGATIVE CONDITIONS OF MAG TAPE FLAG, ERROR FLAG, GO BIT, ILLEGAL FUNCTION, READ-COMPARE ERROR AND EMA OVERFLOW ERROR AS A NO TRANSPORT CONDITION WILL ALLOW.

TEST 11 EXERCISES THE IOT'S SKEF AND SKTD UNDER INTERESTING CONDITIONS AS WELL AS OTHER SKIP AND NO SKIP IOT'S IN THE 672X IOT DECODER.

TEST 12 CHECKS THE UNIQUENESS OF INTERRUPTS AS CAUSED BY MAG TAPE FLAG AND THE ERROR FLAG.

TEST 13 VERIFIES THE UNIQUENESS OF ALL 672X IOT'S IN RELATION TO CLF AND SBRM.

TEST 14 COMPLETELY TESTS THE PROPER OPERATION OF THE READ-COMPARE LOGIC.

12. LISTING (ATTACHED)  
-----

/TM8E CONTROL TEST PART 1 MAINDEC=08-DHTMA-A=L

/COPYRIGHT 1972, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS,

\*\*\*\*\*  
/ WARNING: \*\*\*\*\*  
/ THIS PROGRAM SHOULD NOT BE STOPPED UNLESS \*\*\*\*\*  
/ MONITOR HAS TYPED "C", \*\*\*\*\*  
/ \*\*\*\*\*

/THIS DIAGNOSTIC PROGRAM TESTS AS MUCH OF THE TM8E CONTROL AS  
/IS POSSIBLE WITHOUT UTILIZING A TRANSPORT SYSTEM AND MAY  
/RESIDE IN ANY EXISTING MEMORY FIELD.

/BASIC STARTING PROCEDURE!  
/POWER DOWN THE TRANSPORT SYSTEM AS NECESSARY  
/TO ENSURE ALL SIGNALS RETURNING FROM THE TRANSPORT SYSTEM ARE HIGH,  
/LOAD THE PROGRAM INTO THE DESIRED MEMORY FIELD WITH THE BINARY LOADER;  
/LOAD THE INSTRUCTION AND DATA FIELDS WITH THE PROGRAM FIELD,  
/LOAD ADDRESS 200,  
/CLEAR ALL SWITCHES,  
/SET SR BIT 0 IF IT IS DESIRED TO SELECT TESTS WITH THE MONITOR,  
/IF SR0 IS CLEAR, ALL TESTS WILL BE RUN,  
/SET ANY OTHER SR OPTIONS DESIRED;  
/DEPRESS "CLEAR", THEN "CONTINUE".

/SWITCH REGISTER CONTROL OPTIONS.

/SR BIT STATE CONTROL  
/0! RUN ALL TESTS NORMALLY,  
/1 IF SET TO 1 ANYTIME PRIOR TO THE QUICK  
/ VERIFY MESSAGE, SELECT TESTS WITH MONITOR,  
/ IF SET TO 1 AFTER QUICK VERIFY MESSAGE, RUN QUICK VERIFY,  
/ (1 PASS OF EACH SELECTED TEST);  
/1! TEST PROGRESS REPORTS;  
/ NO TEST PROGRESS REPORTS,  
/2! GO TO MONITOR ON ANY ERROR OCCURRENCE,  
/ GO TO MONITOR ONLY UPON FATAL ERROR,  
/3! PRINT ERROR MESSAGES AND IF "TR" IS PENDING ENABLE TRACE MESSAGES,  
/ PRINT ONLY FATAL ERROR MESSAGES,  
/4! ERROR MESSAGES DISPLAY NECESSARY DATA ONLY,  
/ ERROR MESSAGES DISPLAY ALL DATA WHETHER APPLICABLE  
/ OR NOT, THIS SWITCH SHOULD BE USED ONLY

/ IF TEST 01 PASSES,  
/NOTE: IF SR5=1 WHEN EXITING MONITOR TO RUN A NEW TEST SELECTION, THE  
/PROGRAM MAY HANG UP IN THE ERZ08 TEST, RELEASING THIS SWITCH MOMENTARILY  
/WILL ALLOW THE PROGRAM TO CONTINUE WITH THE NEW SELECTED TESTS,  
/S1 0 NO SUBTEST LOOP,  
/ 1 LOOP ON SUBTEST, (SCOPE LOOP)  
/LOOPS 6-9 ARE SELECTED BY SR 6-9, THESE LOOPS ALLOW THE USER TO LOOP  
/ON A SMALL GROUP OF SUBTESTS WITHIN THE CURRENT TEST, THEREBY INCREASING  
/THE TESTING FREQUENCY OF A SELECTED FUNCTION, THE LISTING MUST BE REFERENCED  
/TO DETERMINE WHICH SUBTESTS ARE INCLUDED IN EACH LOOP,  
/IF LOOP N IS SELECTED, SETTING LOOP N=1 SWITCH WILL TRANSFER TO THAT  
/LOOP UPON COMPLETION OF THE PRESENT LOOP, BY THIS MEANS IT IS POSSIBLE

/TO BACKTRACK IN A GIVEN TEST,

/6-9:	0	NO LOOP 6-9 RESPECTIVELY,
/	1	LOOP 6-9 RESPECTIVELY, (LOOP ON SMALL GROUP
/		OF SUBTESTS),
/10:	0	NO LOOP ON CURRENT TEST (T01-T14),
/	1	LOOP ON CURRENT TEST (T01-T14),
/11:	0	NO LOOP ON COMPLETE TEST AS SELECTED,
/	1	LOOP ON COMPLETE TEST AS SELECTED
/		(OCTAL PASS NUMBER TYPED AT END OF EACH PASS),

/TTY KEYBOARD COMMAND CONTROL.  
 /NOTES: 1. THE FOLLOWING COMMANDS MAY BE ENTERED WHEN "C" IS  
 /TYPED.  
 / (UPON ENTRY MONITOR TYPES THE CONTENTS OF LOCATION 2, AND THE TEST  
 / THAT IS CURRENTLY BEING RUN, THEN "C")  
 /2. "←" INDICATES "RETURN" KEY; "RUSOUT" ABRPTS A COMMAND STRING  
 / PRIOR TO "←"; "SPACE" MAY BE INPUT AT ANY POINT.  
 /3. "ALTMODE" INTERRUPTS THE MAIN PROGRAM AND RETURNS TO MONITOR  
 / ANYTIME THAT THE INTERRUPT INDICATOR IS ON.

/COMMAND RESULT  
 /"T01T02T05TNN←" RUN THE TESTS INDICATED. IF A TEST NUMBER IS INPUT 2  
 / TIMES, THAT TEST IS NOT RUN.  
 / RUN ALL TESTS.  
 /"TA←" CONTINUE FROM POINT OF INTERRUPTION, IF NO TESTS ARE  
 /"C←" PENDING OR A FATAL ERROR OCCURS, "C←" RESULTS IN  
 / SYNTAX ERROR.  
 /"EX←" IF A SUBTEST CONTAINS A LOOP TO CHECK OUT VARYING DATA  
 / PATTERNS, AND IF ERRORS KEEP OCCURRING PREVENTING CONTINUING  
 / ON QUICKLY WITH THE NEXT SUBTEST, DEPRESS "ALTMODE"  
 / (OR ALLOW RETURN TO MONITOR ON ERROR IF THE INTERRUPT SYSTEM  
 / IS OFF DURING THIS TEST.)  
 / THEN "EX←". THIS WILL EXIT THE CURRENT SUBTEST AFTER  
 / THE NEXT ERROR ALLOWING THE TEST TO CONTINUE AT THE NEXT  
 / SUBTEST. THE "EXITFL" WHICH IS SET BY "EX←"  
 / IS CLEARED AS SOON AS IT IS UTILIZED FOR AN  
 / ESCAPE. IF NO TESTS ARE PENDING OR A FATAL ERROR  
 / OCCURS, "EX←" RESULTS IN SYNTAX ERROR.  
 / TRACE USING ERROR MESSAGE FORMAT IF ERROR DOES NOT OCCUR.  
 /"TR" "TR" MAY PRECEED "TA", "TNN", "C", OR "EX" COMMANDS;  
 / TRACE REMAINS IN EFFECT UNTIL "C" IS DISPLAYED AGAIN.  
 / IT RESULTS IN A MODIFIED ERROR MESSAGE BEING DISPLAYED  
 / FOR EACH SUBTEST RUN IF THAT TEST PASSES; THESE  
 / PRINTOUTS ARE CONTROLLED BY THE SAME SR OPTIONS AS  
 / ERROR MESSAGES.  
 /"DN:MMMM" DISPLAY THE CONTENTS OF MEMORY LOCATION MMMM IN FIELD N.  
 / IF A LINE FEED IS TYPED THE NEXT SEQUENTIAL MEMORY  
 / LOCATION IS TYPED, WHEN THE ADDRESS OVERFLOWS THE  
 / NEXT MEMORY FIELD WILL BE REFERENCED AND THE  
 / CONTENTS DISPLAYED. A CARRIAGE RETURN ALLOWS THE INPUT  
 / OF A NEW COMMAND.



/ERRORS:

/IF AN ERROR OCCURS, REFER TO THE ERROR PRINTOUT AND THE PROGRAM LISTING FOR THE SUBTEST FAILURE DESCRIPTION.

/ERROR PRINTOUTS:

/HEADER  
/\*\*\*\*\*  
DEFINITION  
\*\*\*\*\*

```

/*ERNX      ERROR OCCURRED IN TEST NN SUBTEST X(TNNX);
/*FENN      SAME AS ABOVE EXCEPT FATAL ERROR.
/*TR*NNX    TRACE PRINTOUT (FROM "TR" MONITOR COMMAND);
/*PC:NNNN   ADDRESS IN PROGRAM AT WHICH ERROR OCCURRED;
/*11:NNNN   OCTAL CODE FOR 1011;
/*12:NNNN   OCTAL CODE FOR 1012;
/*60:NNNN   GOOD TEST VALUE; DESCRIBED IN SUBTEST LISTING;
/*80:NNNN   BAD OR ACTUAL TEST VALUE; DESCRIBED IN SUBTEST LISTING;
/*00:NNNN   USUALLY PREVIOUS GOOD; DESCRIBED IN SUBTEST LISTING;
/*THE FOLLOWING DISPLAY THE CONTENTS OF THE INDICATED REGISTER:
/*C:NNNN    WORD COUNT;
/*CA:NNNN   CURRENT ADDRESS.
/*CM:NNNN   COMMAND.
/*FS:NNNN   FUNCTION * STATUS;
/*MS:NNNN   MAIN STATUS;
/*DB:NNNN   DATA BUFFER;
/*AC:NNNN   ACCUMULATOR (USED FOR SKIP ERRORS ONLY).

```

/PROGRAM DIRECTORY:

NO PUNCH  
#7600

7600

/REFER TO ASSEMBLY COLUMN JUST TO THE LEFT OF THE MNEMONIC FOR THE  
/ABSOLUTE STARTING ADDRESS OF THAT PARTICULAR PROGRAM SEGMENT,  
/(THIS IS FOR REFERENCE ONLY. DO NOT USE THESE STARTING ADDRESSES.)

/MNEMONIC - PROGRAM SEGMENT.

/TEST SECTION:  
/670X,671X IOT EXISTENCE AND BASIC WC,CA,CM,DB,FR REGISTER TEST.  
/WC,CA,CM,DB,FR REGISTER DATA TEST.  
/BASIC FUNCTIONS EXISTENCE TEST.  
/WC,CA REGISTERS INCREMENT TEST (INCLUDES CM6=8).  
/MEMORY-DB DATA TEST.  
/CA REGISTER MEMORY REFERENCE TEST (USES ALL EXISTING MEMORY FIELDS).  
/STATUS REGISTERS STAND ALONE TEST.  
/BASIC TESTS FOR MTF, ILLEGAL FUNCTION, READ COMPARE, AND EF.  
/MTF AND EF SKIP AND SKIP TEST.  
/MTF-EF INTERRUPT TEST.  
/672X IOT UNIQUENESS TEST.  
/COMPLETE READ COMPARE TEST.

/PROGRAM CONTROL SECTION:

EXEC  
MONIT  
INTSEV

7614 4200  
7615 4400  
7616 5400

ERRORS  
COMP  
FE00A

7617 5000  
7620 5600  
7621 6677

/ERROR HANDLERS:  
/ERROR HANDLING ROUTINE.  
/ERROR DETECTOR.  
/ERROR TABLE.

/UTILITIES:

/TM8E IOT SUBROUTINES,  
/(IF THE DEVICE CODES ARE  
/NOT 70-72 IN YOUR SYSTEM  
/CHANGE ONLY THE IOT CODES IN  
/EACH OF THE 23 SUBROUTINES AND IN THE LOCATION MARK3.)  
/LOOP SENSORS,  
/PACKED TEXT MESSAGE GENERATOR,  
/TTY ROUTINES,  
/OCTAL NUMBER OUTPUT EDITOR,  
/"INITIALIZE" GENERATORS,  
/FUNCTION ROUTINES,

S6700  
MARK3

7622 4000  
7623 1624

LAS5  
AMG8E  
TYPE  
EDIT  
CLR1  
OFFLINE

7624 6105  
7625 6200  
7626 6244  
7627 6301  
7630 6400  
7631 6432

/DATA SECTION:

/TEST01 IOT LISTS,  
/TEST03 FUNCTION LIST,  
/TEST11 IOT LIST  
/TEST12 INTERRUPT ENABLE LIST,  
/TEST13 IOT LIST,  
/CONSTANTS, POINTERS, ETC.  
/MESSAGE TEXTS.

7632 6613  
7633 6647  
7634 3146  
7635 3337  
7636 3146  
7637 0000  
7640 7163

ENPUNCH

```

0000 *0000
0000 0000
0001 5001
0002 3002
0003 0003
0004 1200
0005 2003

      0
      JMP      1
      2
      3
T3LODP, T3LOAD
K3,      3
    
```

/LOCATIONS 1-3 ARE ASSEMBLED SET  
/UP FOR HIGH SPEED DUMP. THEY  
/ARE CHANGED FOR INTERRUPT HANDLING  
/BY THE PROGRAM,

```

0020 *0020
      /WIDE USAGE NUMERIC CONSTANTS, (POSITIVE)
0021 K1,      1
0022 K2,      2
0023 K7,      7
0024 K10,     10
0025 K20,     20
0026 K40,     40
0027 K70,     70
0028 K100,    100
0029 K200,    200
0030 K215,    215
0031 K400,    400
0032 K2000,  2000
0033 K4000,  4000
0034 K4040,  4040
0035 K7400,  7400
0036
    
```

```

0037 /WIDE USAGE NUMERIC CONSTANTS, (NEGATIVE)
0038 M2,     -2
0039 M4,     -4
0040 M5,     -5
0041 M14,   -14
0042 M20,   -20
0043 M260,  -260
0044 M270,  -270
0045
    
```

```

0046 /WIDE USAGE INSTRUCTIONAL CONSTANTS:
      KCDF,   CDF
    
```

```

0047 /TRIAL BUFFER AREA.
0048 XBUFF, XBUFF
0049 XBUFF, 0
0050
0051
0052
0053
0054
    
```

```

0055 /WIDE USAGE POINTERS
0056 EXEC, EXEC
0057 FUNC, FUNC
0058 SCDF, SCDF
0059 INSRP, INSRP
    
```

0061 6200 AMG8EP, AMG8E  
 0062 4400 MONITP, MONIT  
 0063 6301 EDITP, EDIT  
 0064 6326 EDTEMP, EDTEM  
 0065 6262 BELLP, BELL  
 0066 6270 CRLF, CRLF  
 0067 7313 ERMSGP, ERMSG  
 0070 6244 TYPEP, TYPE  
 0071 5507 EXECFP, EXECFX

/TM8E IOT SUBROUTINE POINTERS,

0072 4000 S6700P, S6700  
 0073 4004 SLWCRP, SLWCR  
 0074 4010 SCWCRP, SCWCR  
 0075 4014 SLCARP, SLCAR  
 0076 4020 SCCARP, SCCAR  
 0077 4024 SLCMRP, SLCMR  
 0100 4030 SLFGRP, SLFGR  
 0101 4034 SLDBRP, SLDBR  
 0102 4040 S6710P, S6710  
 0103 4044 SRWCRP, SRWCR  
 0104 4050 SRCARP, SRCAR  
 0105 4054 SRMSRP, SRMSR  
 0106 4060 SRFMRP, SRFMR  
 0107 4064 SRDBRP, SRDBR  
 0110 4070 S6720P, S6720  
 0111 4074 SSKEFP, SSKEF  
 0112 4100 SSKCBP, SSKCB  
 0113 4105 SSKTDP, SSKTD  
 0114 4112 SSKTRP, SSKTR  
 0115 4117 SCLFP, SCLF  
 0116 4124 SCLTP, SCLT  
 0117 4130 SSDLEP, SSDLE  
 0120 4134 SSBMP, SSBRM  
 0121 4140

/LOAD IOT,

/LWCR  
 /CWCR  
 /LCAR  
 /CCAR  
 /LCMR  
 /LFGR  
 /LDBR  
 /READ IOT,  
 /RWCR  
 /RCAR  
 /RMSR  
 /RCMR  
 /RFSR  
 /RDBR  
 /CONTROL IOT,  
 /SKEF  
 /SKCB  
 /SKTD  
 /SKTR  
 /CLF  
 /CLT  
 /SDLE  
 /SBRM

/PSEUDO MNEMONIC SUBROUTINE POINTERS,

0122 6547 /PBSWP, SRBSW  
 0123 6400 CLR1P, CLR1  
 0124 6406 CLR2P, CLR2  
 0125 6414 CLR3P, CLR3  
 0126 6105 LAS5P, LAS5  
 0127 6113 LAS6P, LAS6  
 0130 6121 LAS7P, LAS7  
 0131 6127 LAS8P, LAS8  
 0132 6135 LAS9P, LAS9  
 0133 5600 COMPP, COMP  
 0134 5000 ERRORP, ERRORS  
 0135 6534 DATBRP, DATBRK  
 0136 6525 CMDP, CMD  
 0137 6432 OFFLNP, OFFLINE  
 0140 6436 REWNP, REWIND  
 0141 6442 READDP, READ  
 0142 6446 RDCMP, RDCMP  
 0143 6452 WRITP, WRIT

/BSW  
 /CLEAR1  
 /CLEAR2  
 /CLEAR3  
 /LOOPS  
 /LOOP6  
 /LOOP7  
 /LOOP8  
 /LOOPY  
 /COMPAR  
 /BREAK  
 /COMAND  
 /OFFLIN  
 /REWIND  
 /READ  
 /RDCOMP  
 /WRITE

```

0144 6496 EOFF, EOFF
0145 6462 SPFORP, SPFORP
0146 6466 SPREV, SPREV
0147 6472 XGAPP, XGAPP
0150 6500 GOEP, GOE
0151 0565 SIOT1P, SIOT1
0152 0570 SIOT2P, SIOT2
0153 6541 WCOVER, WCOVER

/WIDE USAGE TEMPORARIES;
GOOD, 0
BAD, 0
OLD, 0
TXXTM1, 0
TXXTM2, 0
TXXTM3, 0
SAVEAC, 0
SAVEL, 0
INTEF, 0
PRGFLD, 0
INTMTF, 0
MEMFLD, 0
ACLOC, 0

/WIDE USAGE SOFTWARE FLAGS AND INDICATORS;
TRACE, 0
INTFLG, 0

TTOFLG, 0
EXITFL, 0
ACTFLG, 0
TSTAT, 0

TSTNUM, 0

/WIDE USAGE TEMPORARIES;
/GOOD DATA;
/BAD DATA;
/OLD DATA;
/THREE TEMPORARIES FOR
/FORMAL TEST USAGE ONLY;

/AC SAVED;
/LINK SAVED;
/IF INTERRUPT SERVICE ADDRESS OCCUPIES;
/MEMORY FIELD WHICH PROGRAM OCCUPIES;
/MTTF INTERRUPT SERVICE ADDRESS;
/NUMBER OF EXTENDED MEM FIELDS PRESENT;
/CONTENTS OF AC FOR ILLEGAL SKIP;

```

```

0171 0000
0172 0000

0173 0000
0174 0000
0175 0000
0176 0000

0177 0000

/THIS IS WHERE THE PROGRAM STARTS;
/*****
*200
START, JMP I EXECR /START HERE, GO TO EXEC,
0201 STL
0202 JMP I EXECR

```

/TEST SECTION,

/THE TEST SECTION IS COMPOSED OF 12 TESTS (T01-T14) EACH  
/OF WHICH IS MADE UP OF SUBTESTS DESIGNATED BY THE LETTERS A-Z, THEN 1-9.

/THE SUBTEST FORMAT IS DESCRIBED BELOW,

/SUBTEST FORMAT FOR NON-DATA TESTS:

```

/T01Z, INST /HOUSEKEEP TO SET UP LOOPS AND SPECIAL LOCATIONS
/ INST /SUCH AS "GOOD" AND "OLD",
/ RUN TEST
/ AT THE END OF THE TEST, AC CONTAINS SOME PREDETERMINED VALUE,
/ COMPAR /COMPARE THE VALUE IN AC WITH THE VALUE IN "GOOD",
/ /IF AC="GOOD" CHECK FOR SUBTEST LOOP,
/ /IF AC NOT EQUAL TO "GOOD" GO TO "ERRORS", TAKE
/ THE REQUIRED ACTION, THEN CHECK FOR SUBTEST LOOP,
/ ER01Z /ADDRESS OF ERROR STATUS WORDS,
/ JMP T01Z /SUBTEST LOOP JUMP-REENTER HERE IF
/ /SUBTEST LOOP REQUESTED,
/ ETC. /POINT OF CONTINUATION,

```

/THE SUBTEST FORMAT FOR DATA TESTS HAS ONE ADDITIONAL  
/INSTRUCTION FOLLOWING THE SUBTEST LOOP JUMP:

```

/ JMP T01Z /SUBTEST LOOP JUMP,
/ JMP TAG /THIS INSTRUCTION RETURNS TO USE NEW DATA,
/IF AN "EX-" COMMAND IS PENDING AND AN ERROR OCCURS, THE ERROR
/ROUTINE ("ERRORS") WILL SKIP THIS INSTRUCTION THEREBY INHIBITING ANY
/MORE DATA FROM BEING USED IN THIS SUBTEST, THIS ENABLES THE USER
/TO ESCAPE FROM A FAILING DATA TEST AND PROCEED TO THE NEXT SUBTEST,

```

/TEST 01, 670X, 671X IOT EXISTENCE AND BASIC NON-STATUS REGISTERS /TEST.

/1, VERIFIES THAT 670X AND 671X IOT'S EXIST EXCEPT FOR /RMSR AND THAT WC, CA, CM, CB AND PR0=4 CAN BE /LOADED AND READ WITH ALL 1'S AND ALL 0'S, THE FACT /THAT EACH OF THESE IOT'S FUNCTION ONLY WITH THEIR /RELATED REGISTERS AND NOT WITH ANOTHER REGISTER IS /ALSO VERIFIED.

/2, TESTS T010-T01P ARE GENERALIZED AND ARE CONTROLLED BY /A LOCAL EXECUTIVE WHICH SELECTS THE IOT'S TO USE.

/3, LOOPS 6-9:  
/3.1 LOOP 6 CYCLES ON T01A-T01C (AC CLEARING)  
/3.2 LOOP 7 CYCLES ON T010-T01N (IOT EXISTENCE & BASIC REGISTER)  
/3.3 LOOP 8 CYCLES ON T010-T01P (IOT UNIQUENESS)

0203 0000 TEST01, 0 IAC  
0204 7001 DCA  
0205 3177 TSTNUM  
0206 6002 IOF INTFLG  
0207 3172 DCA GOOD  
0210 3154 DCA

/RUN WITH INTERRUPT SYSTEM OFF.

0211 7240 /VERIFY 6700 CLEARS AC,  
0212 4472 T01A, CLA CMA  
0213 4533 I6700  
0214 6705 COMPAR  
0215 5211 ER01A  
/IOT 6700 DID NOT CLEAR AC, GD=GOOD AC/ BD=REAL AC,  
T01A JMP

0216 7240 /VERIFY 6710 CLEARS AC,  
0217 4502 T01B, CLA CMA  
0220 4533 I6710  
0221 6707 COMPAR  
0222 5216 ER01B  
/IOT 6710 DID NOT CLEAR AC, GD=GOOD AC/ BD=REAL AC,  
T01B JMP

0223 7240 /VERIFY 6720 DOES NOT CLEAR AC  
0224 3154 T01C, CLA CMA  
0225 7240 DCA GOOD  
0226 4511 CLA CMA  
0227 4533 I6720  
0230 6711 COMPAR  
ER01C  
/IOT 6720 CLEARED AC, GD=GOOD AC, BD=REAL AC,  
T01C+2 JMP

0231 5225 LOOP6  
0232 4527 JMP  
0233 5210 T01A-1 /\*\*\*\*\*LOOP 6\*\*\*\*\*

```

0234 4775 JMS I T1EX1P /TO EXECUTIVE,
/VERIFY LOAD IOT LXXR CLEARS AC.
T01D, DCA GOOD
0235 3154 TAD TXXTM1
0236 1157 LXXR
0237 4551 COMPAR
0240 4533 ER01D
0241 6713 /LOAD IOT LXXR DID NOT CLEAR AC, GD=GOOD AC; BD=REAL AC;
/11=LXXR, JMP T01D+1 /SUBTEST LOOP,
0242 5236
/VERIFY THAT LXXR WITH AC=7777, THEN RXXR RESULTS IN A
/NON-ZERO AC, TXXTM1
T01E, TAD TXXTM1
LXXR
0243 1157 /CLEAR AC TO ENSURE GOOD TEST,
0244 4551 CLA
0245 7200 RXXR
0246 4552 AND /MASK BITS 0=4 FOR FS ONLY,
0247 0157 SNA CLA
0250 7650 CMA
0251 7040 COMPAR
0252 4533 ER01E
0253 6715 /LXXR WITH AC=7777, THEN RXXR DID NOT RESULT IN A NON-ZERO
/AC, I1=LXXR; I2=RXXR /SUBTEST LOOP
JMP T01E
0254 5243
/VERIFY XX REGISTER TO BE LOADED AND READ ALL 1'S BY LXXR
/T01F, CLA CMA GOOD
0255 7240 DCA TXXTM1
0256 3154 TAD
0257 1157 LXXR
0260 4551 CLA
0261 7200 RXXR
0262 4552 AND /AC CLEARED TO ENSURE GOOD TEST,
0263 0157 TXXTM1 /MASK TO BITS 0=4 FOR FS ONLY,
0264 1160 TAD TXXTM2 /ADD IN COMPLEMENT MASK,
0265 4533 COMPAR
0266 6717 ER01F
/XX REGISTER NOT LOADED AND/OR READ WITH ALL 1'S USING
/LXXR AND RXXR, GD=GOOD AC; BD=REAL AC; I1=LXXR;
/I2=RXXR, JMP T01F+2 /SUBTEST LOOP
0267 5237
/VERIFY THAT LXXR WITH AC=0000, THEN RXXR RESULTS IN
/AT LEAST ONE BIT CLEAR IN AC.
T01G, DCA GOOD
LXXR
0270 3154 CLA
0271 4551 RXXR
0272 7200 AND
0273 4552 TXXTM1 /MASK BITS 0=4 FOR FS ONLY,
0274 0157 TAD TXXTM2 /ADD IN COMPLEMENT MASK,
0275 1160 IAC /ADD 1 TO CHECK FLIP TO 0,
0276 7001 SNA CLA
0277 7650

```



0300 7040  
0301 4533  
0302 6721  
  
0303 5271

CMA  
COMPAR  
ER01C  
/LXXR WITH AC=0000, THEN RXXR DID NOT RESULT IN AT  
/LEAST ONE BIT CLEAR IN AC; I1=LXXR; I2=RXXR;  
T01G+1  
JMP

0304 4551  
0305 7200  
0306 4552  
0307 0157  
0310 4533  
0311 6723

/VERIFY XX REGISTER CAN BE LOADED AND READ ALL 0'S;  
T01H,  
LXXR  
CLA  
RXXR  
AND  
TXXTM1 /MASK BITS 2-4 FOR FS ONLY  
COMPAR  
ER01H  
/LXXR WITH AC=0000, THEN RXXR DID NOT RESULT IN A ZERO AC;  
/GD=GOOD AC; BD=REAL AC; I1=LXXR; I2=RXXR;  
JMP T01H

0312 5304  
  
0313 1157  
0314 4551  
0315 4525  
0316 4552  
0317 0157  
0320 4533  
0321 6725

/VERIFY CLT CLEARS THE XX REGISTER WHICH IS PRELOADED TO 1'S  
/BY LXXR; LOBR WITH AC=0000 USED FOR DB,  
T01I,  
TAD  
LXXR  
CLEAR3  
RXXR  
AND  
TXXTM1 /MASK BITS 0-4 FOR FS ONLY  
COMPAR  
ER01I

0322 5313  
  
0323 4525  
0324 7240  
0325 4552  
0326 0157  
0327 4533  
0330 6727  
0331 5323

/XX REGISTER NOT CLEARED BY CLT (OR LOBR IF DB); GD=GOOD AC; BD=REAL AC;  
/I1=LXXR; I2=RXXR;  
JMP T01I  
/SUBTEST LOOP,  
  
/VERIFY RXXR CLEARS AC. REGISTER PRELOADED TO 0000;  
T01J,  
CLEAR3  
CLA CMA  
RXXR  
AND  
TXXTM1 /MASK BITS 0-4 FOR FS ONLY  
COMPAR  
ER01J  
/RXXR DOES NOT CLEAR AC. GD=GOOD AC; BD=REAL AC; I2=RXXR;  
JMP T01J  
/SUBTEST LOOP

0332 1157  
0333 4551  
0334 7200  
0335 4776

TAD  
LXXR  
CLA  
JMS I T1EX2P  
/PRELOAD REGISTER TO ALL 1'S  
/FOR CXXR CHECKS  
/ONLY TEST REGISTERS FOR WHICH  
/CXXR APPLIES;

0336 7200  
0337 4551  
0340 4552  
0341 7001  
0342 7650

/VERIFY CXXR CLEARS AT LEAST ONE BIT IN THE REGISTER WHICH IS  
/PRELOADED TO ALL 1'S,  
T01K,  
CLA  
CXXR  
RXXR  
IAC  
SNA CLA

```

0343 7040      CMA
0344 4533      COMPAR
0345 6731      ER01K
/CXXR DID NOT CLEAR AT LEAST ONE BIT IN XX REGISTER
/I1=CXXR; I2=RXRX      JMP      T01K+1      /SUBTEST LOOP

0346 5337

0347 7200      /VERIFY THAT CXXR CLEARS THE XX REGISTER
0350 4551      T01L,      CLA
0351 4552      CXXR
0352 4533      RXXR
0353 6733      COMPAR
ER01L
/CXXR DID NOT CLEAR XX REGISTER, GD=GOOD AC; BD=REAL AC;
/I1=CXXR; I2=RXRX,      JMP      T01L+1      /SUBTEST LOOP

0354 5350

0355 7240      /VERIFY CXXR CLEARS AC.
0356 4551      T01M,      CLA CMA
0357 4533      CXXR
0360 6735      COMPAR
ER01M
/CXXR DOES NOT CLEAR AC. GD=GOOD AC; BD=REAL AC;
/I1=CXXR,      JMP      T01M      /SUBTEST LOOP

0361 5355

0362 7240      /VERIFY CXXR WITH AC=7777 DOES NOT SET ANY BIT IN XX
0363 4551      /REGISTER.
0364 4552      T01N,      CLA CMA
0365 4533      CXXR
0366 6737      RXXR
COMPAR
ER01N
/CXXR SET AT LEAST ONE BIT IN XX REGISTER, GD=GOOD AC;
/BD=REAL AC; I1=CXXR; I2=RXRX,      JMP      T01N      /SUBTEST LOOP

0367 5362

0370 4777      JMS I      T1EX3P      /BACK TO EXECUTIVE;

0371 4530      T1LP7,      LOOP7
0372 5232      JMP      T01D-3      /*****LOOP 7****

0373 5774      JMP I      ,*1      /GO TO T010 WHEN D=0N DONE
0374 0400      T010L
0375 0450      T1EX1P, T1EX1
0376 0473      T1EX2P, T1EX2
0377 0501      T1EX3P, T1EX3

0400      PAGE

0400 4305      T010L,      JMS      T1EX4      /TO EXECUTIVE TO INSERT
0401 4525      CLEAR3      /IOT'S.
0402 3154      DCA      GOOD

```

/VERIFY SPECIFIC LOAD, CLEAR, AND READ IOT'S AFFECT ONLY THEIR  
 /RELATED REGISTERS, AC SET TO 7777 PRIOR TO LOAD OR CLEAR  
 /WITH ALL OTHER REGISTERS CLEAR, IOT'S INSERTED BY EXECUTIVE,  
 T010,

0403 7240  
 0404 4551  
 0405 4552  
 0406 0157  
 0407 4533  
 0410 6741

CLA CMA  
 LXXR  
 RXXR  
 AND TXXTM1  
 COMPAR  
 ER010

/IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS  
 /REGISTER LOADED BY LXXR, GD=500 AC; BD=REAL AC;  
 /I1=LXXR; I2=RXXR,

0411 5203

JMP T010 /SUBTEST LOOP

/VERIFY THAT SPECIFIC LOAD, CLEAR AND READ IOT'S AFFECT  
 /ONLY THEIR RELATED REGISTERS, AC SET TO 0000 PRIOR TO  
 /LOAD OR CLEAR WITH ALL OTHER REGISTERS SET TO 1/S;  
 /SAME METHOD AS T010,

0412 4525  
 0413 7240  
 0414 3154  
 0415 7240  
 0416 4473  
 0417 7240  
 0420 4475  
 0421 7240  
 0422 4477  
 0423 7240  
 0424 4501  
 0425 1374  
 0426 4500  
 0427 7200  
 0430 4551  
 0431 4552  
 0432 0157  
 0433 1160  
 0434 4533  
 0435 6743

CLEAR3  
 CLA CMA GOOD  
 DCA CMA  
 CLA CMA  
 LMC R  
 CLA CMA  
 LCA R  
 CLA CMA  
 LCM R  
 CLA CMA  
 LCB R  
 TAD K7600  
 LFC R  
 CLA  
 LXXR  
 RXXR  
 AND  
 TAD  
 COMPAR  
 ER01P

T01P,

/LOAD ONE REGISTER  
 /READ ANOTHER  
 /MASK BITS 0=4 FOR FS ONLY,  
 /ADD COMPLEMENT MASK,

/IOT LXXR AFFECTS REGISTER READ BY RXXR, OR RXXR READS  
 /REGISTER LOADED BY LXXR, GD=500 AC; BD=REAL AC;  
 /I1=LXXR; I2=RXXR,

0436 5230

JMP T01P+1 /SUBTEST LOOP

0437 4340

JMS T1EX5 /TO EXECUTIVE

0440 4531  
 0441 5773

LOOP8  
 JMP I T1LP7P /\*\*\*\*\*LOOP 8\*\*\*\*

0442 4523  
 0443 1647  
 0444 3246  
 0445 5646  
 0446 0000  
 0447 0203

CLEAR1  
 TAD I  
 DCA  
 JMP I  
 0  
 TEST01  
 /CLEAR REGISTERS = ION;  
 /EXIT,

/TEST 01 LOCAL EXECUTIVE TO SERVICE TESTS T01D=T01N,

T1EX1, 0 TAD T1LS1P  
 0450 0000 DCA 10  
 0451 1356 TAD T1LS2P  
 0452 3010 DCA 11  
 0453 1357 TAD T1LS3P  
 0454 3011 DCA 12  
 0455 1360 TAD M5  
 0456 3012 DCA T1EXT1  
 0457 1041 TAD I 10  
 0460 3361 DCA IOT1  
 0461 1410 DCA I 12  
 0462 3366 TAD IOT2  
 0463 1412 DCA TXXTM1  
 0464 3371 TAD TXXTM1  
 0465 1412 DCA TXXTM2  
 0466 3157 TAD T1EX1  
 0467 1157 CMA  
 0470 7040 DCA  
 0471 3160 JMP I  
 0472 5650 TAD I 11  
 0473 0000 SNA  
 0474 1411 JMP  
 0475 7450 DCA  
 0476 5302 JMP IOT1  
 0477 3366 DCA T1EX2  
 0500 5673 JMP I  
 0501 0000 ISE T1EXT1  
 0502 2361 JMP T1EX1A  
 0503 5261 JMP I  
 0504 5701 JMP I T1EX3

/ENTER PRIOR TO T01D;  
 /SET UP LIST POINTERS

/SET IOT1 FOR LXXR;

/SET IOT2 FOR RXXR;

/SET MASK AND COMPLEMENT  
 /MASK;

/GO BACK AND RUN T01D=T01J;  
 /ENTER AFTER T01J;  
 /SEE IF CXXR APPLIES;

/CXXR APPLIES, PUT IN IOT1,  
 /GO BACK AND RUN T01K=T01N,  
 /ENTER AFTER T01N,  
 /ALL IOT'S CHECKED?  
 /NO, RESET IOT'S  
 /YES, EXIT TO END OF T01N;

/TEST 01 LOCAL EXECUTIVE TO SERVICE TESTS T010=T01P,

T1EX4, 0 TAD T1LS1P  
 0505 0000 DCA 10  
 0506 1356 TAD T1LS2P  
 0507 3010 DCA 11  
 0510 1357 TAD T1LS3P  
 0511 3011 DCA T1EXT1  
 0512 1360 TAD M5  
 0513 3361 DCA T1EXT2  
 0514 1041 TAD I 10  
 0515 3362 DCA IOT1  
 0516 1410 DCA M2  
 0517 3366 TAD T1EXT4  
 0520 1037 DCA T1EXT1  
 0521 3364 DCA ISE  
 0522 2361 ISE T1EXT1  
 0523 2361 TAD T1EXT1  
 0524 1361 DCA I 12  
 0525 3012 DCA M4  
 26 TAD

/SET LIST POINTERS;

/SET FOR 5 LOAD, CLEAR COMBINATIONS,

/GET LOAD,  
 /PUT IN IOT1,  
 /SET FOR PAIR COUNTER,

/UPDATE READ POINTER;

/GET HEAD POINTER AND  
 /STORE IN AUTO-INDEX,  
 /SET FOR 4 READS.

```

0527 3363          DCA      T1EXT3
0530 1412          TAD I    12
0531 3371          DCA      IOT2
0532 1412          TAD I    12
0533 3157          DCA      TXXTM1
0534 1157          TAD      TXXTM1
0535 7040          CMA
0536 3160          DCA      TXXTM2
0537 5705          JMP I    T1EX4
0540 0000          T1EX5, 0
0541 2363          ISZ     T1EXT3
0542 5330          JMP     T1EX6
0543 2364          ISZ     T1EXT4
0544 7410          SKP     T1EXT4
0545 5353          JMP     :+6
0546 1411          TAD I    11
0547 7450          SNA
0550 5353          JMP     :+3
0551 3366          DCA      IOT1
0552 5324          JMP     T1EX6
0553 2362          ISZ     T1EXT2
0554 5316          JMP     T1EX7
0555 5740          JMP I    T1EX5
0556 6612          T1LS1P, T1LS1-1
0557 6617          T1LS2P, T1LS2-1
0560 6624          T1LS3P, T1LS3-1
0561 0000          T1EXT1, 0
0562 0000          T1EXT2, 0
0563 0000          T1EXT3, 0
0564 0000          T1EXT4, 0
0565 0000          SIOT1, 0
0566 0000          IOT1, 0
0567 5765          JMP I    SIOT1
0570 0000          SIOT2, 0
0571 0000          IOT2, 0
0572 5770          T1LP7P, T1LP7
0573 0371          K7600, 7600
0574 7600

```

```

/GET READ,
/PUT IN IOT2,
/GET MASK, SAVE MASK,
/AND COMPLEMENT MASK,

```

```

/RUN TEST,

```

```

/ENTER AFTER TEST, 4 READS?
/NO, DO NEXT READ,
/YES, DONE WITH THIS LOAD, CLEAR PAIR?

```

```

/YES, GET CLEAR,
/NO, GET CLEAR,

```

```

/CLEAR APPLIES, RUN TEST AFTER
/SETTING UP READS,
/ALL DONE?
/NO, RUN NEXT PAIR,
/YES, EXIT,

```

```

/IOT1 SUBROUTINE,

```

```

/IOT2 SUBROUTINE,

```

/TEST 02. NON-STATUS REGISTER DATA TEST.

/1. VERIFIES THAT WC, CA, CM, DB AND FR0-4 CAN BE LOADED  
/AND READ USING INCREMENTED COMPLEMENT DATA.

/2. ALL SUBTEST LOOPS CYCLE ON CONSTANT DATA.

/3. LOOPS 6-9:

/3.1 LOOP 6: T02A(WC)

/3.2 LOOP 7: T02B(CA)

/3.3 LOOP 8: T02C(CM)

/3.4 LOOP 9: T02D=T02E(DB,FR0-4)

0600	PAGE	
0600	TEST02, 0	
0601	CLEAR1	/CLEAR ALL-ION;
0602		/VERIFY WC CAN BE LOADED AND READ USING INCREMENTAL
0603		/COMPLEMENTED PATTERN;
0604	T02A,	JMS T02LDS
0605		JMP 7777
0606		JMP T02B-2
0607		TAD TXXTM1
0610		LWCR
0611		CLA
0612		RWCR
		COMPAR
		ER02A
		/WC NOT LOADED OR READ CORRECTLY, GD=GOOD WC; OD=PREVIOUS
0613		/GOOD WC, SUBTEST LOOP CYCLES ON PRESENT DATA;
0614		JMP 106
		JMP T02LDR
0615	LOOP6	
0616	JMP	T02A /*****LOOP 6****
0617		/VERIFY CA CAN BE LOADED AND READ USING INCREMENTAL
0620		/COMPLEMENTED DATA;
0621	T02B,	JMS T02LDS
0622		JMP 7777
0623		JMP T02C-2
0624		TAD TXXTM1
0625		LVAR
0626		CLA
0627		RVAR
		COMPAR
		ER02B
		/CA NOT LOADED OR READ CORRECTLY, GD=GOOD CA; OD=PREVIOUS
0630		/GOOD CA, SUBTEST LOOP CYCLES ON PRESENT DATA;
0631		JMP 106
		JMP T02LDR
		/CLEAR AC TO ENSURE GOOD TEST,

```

0632 4530 LOOP7
0633 5215 JMP T02B-2 /*****LOOP 7*****/

0634 3172 /VERIFY CM CAN BE LOADED AND READ USING INCREMENTAL
0635 6202 /COMPLEMENTED DATA.
0636 4307 T02C, DCA INTFLG /ALLOW NO INTERRUPTS.
0637 7777 IOF T02LDS
0640 5251 JMS 7777
0641 1157 JMP T02D-3
0642 4477 TAD TXXTM1
0643 7200 LCMR
0644 4506 CLA
0645 4533 RCMR
0646 6751 COMPAR
ER02C /CLEAR AC TO INSURE GOOD TEST,

0647 5241 /CM NOT LOADED OR READ CORRECTLY, GD=GOOD CM; OD=PREVIOUS
0650 5331 /GOOD CM, SUBTEST LOOP CYCLES ON PRESENT DATA,
JMP :=6 /SUBTEST LOOP
JMP T02LDR

0651 4523 CLEAR1 /CLEAR ALL = 10N.
0652 4531 LOOP8
0653 5232 JMP T02C-2 /*****LOOP 8*****/

0654 4524 /VERIFY DB CAN BE LOADED AND READ USING INCREMENTAL
0655 4307 /COMPLEMENTED DATA.
0656 7777 T02D, CLEAR2 T02LDS
0657 5270 JMS 7777
0660 1157 JMP T02E
0661 4501 TAD TXXTM1
0662 7200 LDBR
0663 4510 CLA
0664 4533 RDBR
0665 6753 COMPAR
ER02D /CLEAR AC TO ENSURE GOOD TEST,

0666 5260 /DB NOT LOADED OR READ CORRECTLY, GD=GOOD DB; OD=PREVIOUS
0667 5331 /GOOD DB, SUBTEST LOOP CYCLES ON PRESENT DATA,
JMP :=6 /SUBTEST LOOP
JMP T02LDR

0670 4307 /VERIFY FS 0=4 CAN BE LOADED AND READ USING INCREMENTAL
0671 7600 /COMPLEMENTED DATA.
0672 5304 T02E, T02LDS
0673 1157 JMS 7600 /MASK FOR BITS 0=4
0674 4500 JMP T02LP9
0675 7200 TAD TXXTM1
0676 4507 LFGR
0677 0271 CLA
0700 4533 RFSR
0701 6755 AND T02E+1 /MASK
COMPAR
ER02E /FS 0=4 NOT LOADED OR READ CORRECTLY, GD=GOOD FS 0=4;

```

```

0702 5273 /BD=REAL FS 0=4 (MASKED); OD=PREVIOUS GOOD FS 0=4,
0703 5331 JMP .7
      JMP T02LDR

```

```

0704 4532 T02LP9, LOOP9
0705 5252 JMP T02D-2 /*****LOOP 9****
0706 5600 JMP I TEST02 /EXIT,

```

/DATA SUPPLIER FOR WC, CA, CM, DB, FR.  
 /STARTING FROM 0 OUTPUTS THE FOLLOWING INCREMENTAL COMPLEMENTED  
 /DATA PATTERN.

/0: 7777,7777,0,0,7777,7777,0,0  
 /1: 7776,7776,1,1,7776,7776,1,1, ETC.

/THE CALLING SEQUENCE IS:  
 /JMS T02LDS  
 /DATA MASK CONSTANT  
 /EXIT  
 /TEST INSTRUCTIONS  
 /ERROR CHECK  
 /JMP T02LDR TO GET NEW DATA.

```

0707 0000 T02LDS, 0
0710 3157 DCA TXXTM1
0711 1021 TAD K2
0712 1307 TAD T02LDS
0713 1360 DCA TXXTM2
0714 1040 TAD M4
0715 3161 DCA TXXTM3
0716 1037 TAD M2
0717 3345 DCA T02LDM
0720 1154 TAD OLD
0721 3156 DCA TXXTM1
0722 1157 TAD CMA
0723 7040 DCA DCA
0724 3157 DCA AND I
0725 1157 TAD DCA
0726 0707 DCA AND I
0727 3154 DCA DCA
0730 5560 JMP I
0731 2345 T02LDR, ISZ
0732 7410 SKP
0733 5337 JMP
0734 1154 TAD
0735 3156 DCA
0736 5560 JMP I
0737 2161 ISZ
0740 5316 JMP
0741 2157 ISZ
0742 5314 JMP
0743 2307 ISZ
0744 5707 JMP I
0745 0000 T02LDM, 0

/ENTER,
/0 FOR A STARTER,
/COMPUTE RETURN EXIT
/AND SAVE,
/SET FOR FOUR COMPLEMENTS,
/SET FOR TWO SENDS OF ONE PATTERN,
/PUT PREVIOUS DATA
/IN OLD,
/SET LAST DATA,
/COMPLEMENT,
/SAVE,
/GET LAST DATA,
/MASK,
/THIS IS WHAT WE SHOULD READ,
/RUN THE TEST,
/THIS PATTERN SENT 2 TIMES,

/YES, PUT OLD DATA IN OLD,
/RUN PATTERN AGAIN,
/COMPLEMENTING DONE 4 TIMES?
/NO, SET UP NEXT DATA,
/YES, INCREMENT DATA,
/SEND OUT NEXT DATA TRAIN,
/ALL DONE, UPDATE RETURN,
/EXIT,

```



```

/TEST 03. BASIC FUNCTIONS EXISTENCE TEST,
/LOCAL EXECUTIVE,
/1. REFER TO SUBTESTS FOR SPECIFIC TEST INFORMATION,
/2. TESTS T03D-T03I ARE GENERALIZED AND ARE CONTROLLED BY A
/3. LOOPS 6-9:
/3.1 LOOP 6 CYCLES ON T03A-T03C,
/3.2 LOOP 7 CYCLES ON T03D-T03E,
/3.3 LOOP 8 CYCLES ON T03F-T03G,
/3.4 LOOP 9 CYCLES ON T03H-T03I,
/4. EACH SUBTEST LOOP CYCLES ON THE FUNCTION IN CURRENT USE,

```

```

1000 PAGE
1001 TEST03, 0 CLEAR1 /CLEAR ALL REGISTERS=ION
1002 TAD XBUFFP
1003 DCA OLD

```

```

1004 DCA GOOD
1005 CLEAR2 /CLEAR REGISTERS
1006 LWCR /0 TO WC
1007 WRITE
1008 GO
1009 RWCR
1010 COMPAR
1011 ER03A
1012
1013

```

```

1014 /VERIFY WRITE-GO DOES NOT CAUSE A DATA BREAK WITH NO
/TRANSPORT READY, WC STARTS AT 0000 AND SHOULD
/REMAIN 0000.
T03A,
T03A+1 /SUBTEST LOOP,

```

```

1015 /VERIFY WRITE CAUSES TRANSFER OUT OF MEMORY, CA REFERENCES
1016 /ADDRESSES XBUFF OR XBUFF+1, CM6=8 REFERENCE THE PROGRAM
1017 /FIELD, AND THAT DB CAN RECEIVE ALL 1'S. (XBUFF AND
1018 /XBUFF+1 CONTAIN 7777,)
1019 DCA CMA
1020 DCA GOOD
1021 JMS I T3LOOP /LOAD UP REGISTERS,
1022 LDBR /0 TO DB
1023 WRITE /1 DATA BREAK
1024 BREAK
1025 R0BR
1026 COMPAR
1027 ER03B

```

```

1028 /GD=GOOD DB1 0D=ADDRESS XBUFF (CA SHOULD EQUAL THIS
1029 /QUANTITY OR ONE MORE=CA INCREMENT IS NOT BEING
1030 /TESTED,)
1031 JMP T03B+2 /SUBTEST LOOP,

```

```

1032 /VERIFY READ CAUSES TRANSFER INTO MEMORY, CA REFERENCES
1033 /ADDRESSES XBUFF OR XBUFF+1, AND THAT DB CAN SEND ALL
1034 /0'S. (XBUFF AND XBUFF+1 CONTAIN 7777 PRIOR TO

```

```

1027 4404 /BREAK,.)
1030 4501 T03C, JMS I T3LOOP /LOAD UP REGISTERS
1031 4541 LDBR /0 TO DB
1032 4535 READ /1 DATA BREAK
1033 1050 BREAK /SUM
1034 1051 TAD XBUFF XBUFF+1
1035 4533 TAD XBUFF+1
1036 6763 COMPAR
ER03C
/0D=GOOD SUM OF THE CONTENTS OF XBUFF AND XBUFF+1 (ONE OF
/1 THE VALUES SHOULD BE 7777, THE OTHER 0000 YIELDING A
/SUM OF 7777); 0D=ACTUAL SUM; 0D=ADDRESS XBUFF
/1 CA SHOULD EQUAL THIS QUANTITY OR 1 MORE = CA INCREMENT
/NOT BEING TESTED.)
1037 5227 JMP T03C /BSUBTEST LOOP

1040 4527 LOOP6
1041 5204 JMP T03A /*****LOOP6*****/

1042 3154 /VERIFY ALL FUNCTIONS EXCEPT READ DO NOT CAUSE TRANSFER
1043 4776 /INTO PROCESSOR MEMORY. XBUFF AND XBUFF+1 CONTAIN
1044 6647 /7777 PRIOR TO BREAK. DB CONTAINS 0000.
1045 7771 DCA GOOD
1046 1051 JMS I T3EX1P /TO EXECUTIVE TO INSERT FUNCTION.
1047 4404 T3LS1 /FUNCTION LIST POINTER
1048 4501 -7 /NUMBER OF FUNCTIONS TO USE.
1049 0000 +3 /WHERE TO PUT FUNCTIONS;
1050 0000 JMS I T3LOOP /LOAD ALL REGISTERS.
1051 0000 LDBR /0 TO DB.
1052 4535 0 /VARIABLE FUNCTION HERE.
1053 1050 BREAK /1 DATA BREAK.
1054 7041 TAD XBUFF XBUFF AND XBUFF+1 SHOULD
1055 1051 CIA /BE EQUAL AND UNCHANGED.
1056 4533 TAD XBUFF+1
1057 6765 COMPAR
ER03D
/0D=GOOD DIFFERENCE OF THE CONTENTS OF XBUFF+1=XBUFF;
/BD=ACTUAL DIFFERENCE; 0D=ADDRESS XBUFF; FS=FUNCTION
/WHICH FAILED. SUBTEST LOOP CYCLES ON CURRENT FUNCTION;
1060 5777 JMP I T3EX32 /SUBTEST LOOP.
1061 4775 JMS I T3EX2P /BACK FOR ANOTHER FUNCTION;

/VERIFY THAT OFFLINE, REWIND, WEOF, SPCFWD, SPCREV AND READ
/DO NOT CAUSE TRANSFER FROM PROCESSOR MEMORY TO DB;
/XBUFF AND XBUFF+1 CONTAIN 7777 PRIOR TO BREAK. DB CONTAINS 0000.
1062 4776 JMS I T3EX1P /TO EXECUTIVE TO GET FUNCTION;
1063 6651 T3LS1+2 /FUNCTION LIST POINTER;
1064 7772 -6 /# OF FUNCTIONS
1065 1070 +3 /WHERE TO INSERT FUNCTION
1066 4404 JMS I T3LOOP /LOAD REGISTERS

1067 4501 LDBR /0 TO DB
1070 0000 0 /VARIABLE FUNCTION
4535 BREAK /1 DATA

```

1072 4510 RBR /DB SHOULD STILL BE 0000,  
 1073 4533 COMPAR  
 1074 6767 ER03E

1075 5777 /GD=GOOD DB; OD=ADDRESS XBUFF; FS=FUNCTION WHICH FAILED  
 1076 4775 JMP I T3EX32 /SUBTEST LOOP  
 JMS I T3EX2P /GET ANOTHER FUNCTION

1077 4530 LOOP7  
 1100 5240 JMP T030-2 /\*\*\*\*\*LOOP7\*\*\*\*\*

1101 7201 /VERIFY SPCFMD, SPREV, READ, RDCOMP AND WRITE CAUSE  
 1102 3154 /WC INCREMENT FROM 0000 TO 0001.  
 1103 4776 T03F, CLA IAC  
 1104 6654 DCA GOOD  
 1105 7773 JMS I T3EX1P /SET VARIABLE FUNCTION;  
 T3LS1+5  
 -5  
 +3  
 JMS I T3LOOP /LOAD REGISTERS  
 LWCR /TO WC  
 0 /FUNCTION INSERTED  
 BREAK / 1 DATA BREAK  
 RWCR  
 COMPAR  
 ER03F

1106 1111 /GD=GOOD WC; FS=FUNCTION WHICH FAILED,  
 1107 4404 JMP I T3EX32 /SUBTEST LOOP  
 1110 4473 JMS I T3EX2P /GET ANOTHER FUNCTION  
 1111 0000  
 1112 4535  
 1113 4503  
 1114 4533  
 1115 6771

1116 5777 /VERIFY READ, RDCOMP, WRITE CAUSE CA INCREMENT FROM  
 1117 4775 /XBUFF TO XBUFF+1  
 T03G, CLA IAC XBUFFP  
 TAD XBUFF  
 DCA GOOD  
 1120 7201 JMS I T3EX1P /GET A FUNCTION.  
 1121 1047 T3LS1-1  
 1122 3154 -3  
 1123 4776 +2  
 1124 6646 JMS I T3LOOP /LOAD REGISTERS  
 1125 7775 0 /FUNCTION INSERTED HERE,  
 1126 1130 BREAK / 1 DATA BREAK,  
 1127 4404 RWCR  
 1130 0000 COMPAR  
 1131 4535 ER03G  
 1132 4504 /GD=GOOD CA; FS=FAILING FUNCTION,  
 1133 4533 JMP I T3EX32 /SUBTEST LOOP  
 1134 6773 JMS I T3EX2P /GET ANOTHER FUNCTION

1135 5777 LOOP8  
 1136 4775 JMP T03F-2 /\*\*\*\*\*LOOP 8\*\*\*\*\*

1137 4531 /VERIFY OFFLINE, REWIND, WEOF DO NOT CAUSE WC INCREMENT,  
 1140 5277 /WC PRESET TO 0000.  
 T03H, DCA GOOD  
 1141 3154 JMS I T3EX1P /GET FUNCTION  
 1142 4776

```

1143 6651 T3LS1+2
1144 7775 -3
1145 1147 .#2
1146 4473 LWCR
1147 0000 0
1150 4535 BREAK
1151 4503 RWCR
1152 4533 COMPAR
1153 6775 ER03H
/GO=GOOD WC; FS=FAILING FUNCTION /SUBTEST LOOP;
JMP I T3EX32 /GET ANOTHER FUNCTION;
JMS I T3EX2P

1156 4776 /VERIFY OFFLINE, REWIND, WE0F; SPCFWD; SPCREV DO NOT
1157 6651 /CAUSE CA INCREMENT, CA PRESET TO 0000,
1160 7773 T03I,
1161 1163 JMS I T3EX1P
1162 4475 T3LS1+2
1163 0000 -5
1164 4535 .#2
1165 4504 LCAR
1166 4533 BREAK
1167 6777 RCAR
/GO=GOOD CA; FS=FAILING FUNCTION /SUBTEST LOOP
JMP I T3EX32 /GET ANOTHER FUNCTION
JMS I T3EX2P

1172 4532 LOOP9
1173 5337 JMP T03H-2
1174 5600 JMP I TEST03

1175 1231 T3EX2P, T3EX2
1176 1213 T3EX1P, T3EX1
1177 1227 T3EX32, T3EX3+2
1200 1200 PAGE

1200 0000 T3LOAD, 0
1201 4524 CLEAR2
1202 7240 CLA CMA
1203 3050 DCA CMA
1204 7240 CLA CMA
1205 3051 DCA
1206 1047 TAD
1207 4475 LCAR
1210 1165 TAD
1211 4477 LCMR
1212 5600 JMP I T3LOAD

1213 0000 T3EX1,
1214 1613 TAD I
1215 3010 DCA
1216 2213 IS#

```

/LOAD XBUFF AND XBUFF+1  
 /WITH 7777, AND LCAR WITH  
 /XBUFF ADDRESS;  
 /TEST 03 FUNCTION SELECTOR EXECUTIVE,  
 /GET CALL #1 \* THIS IS STARTING  
 /FUNCTION IN T3LS1;  
 /GET CALL #2 \* THIS IS THE NUMBER

1217	1613	TAD I	T3EX1	/OF FUNCTIONS TO BE USED,
1220	3157	DCA	TXXTM1	
1221	2213	ISZ	T3EX1	/UPDATE CALL POINTER TO LOCATION IN
1222	1613	TAD I	T3EX1	
1223	3160	DCA	TXXTM2	
1224	2213	ISZ	T3EX1	
1225	1410	TAD I	10	/WHICH FUNCTION TO BE INSERTED AND
1226	3560	DCA I	TXXTM2	/POINT OF EXIT,
1227	4524	CLEAR2		
1230	5613	JMP I	T3EX1	/RUN TEST WITH THIS FUNCTION,
1231	0000	Ø		/ENTER AT END OF SUBTEST,
1232	2157	ISZ	TXXTM1	/ALL FUNCTIONS USED?
1233	5225	JMP	T3EX3	/NO, GET NEXT FUNCTION,
1234	5931	JMP I	T3EX2	/YES, EXIT TO END OF THIS SUBTEST
				/PLUS 1 INSTRUCTION,

/TEST 04, FULL WORD COUNT AND CURRENT ADDRESS INCREMENT TEST  
/INCLUDING CM6-8,

/1. REFER TO SUBTEST FOR SPECIFIC TEST INFORMATION,  
/2. LOOPS 6-7:  
/2.1 LOOP 6 CYCLES ON T04A, (WC INCREMENT)  
/2.2 LOOP 7 CYCLES ON T04B, CD, AND E (CA AND EMA TESTS),  
/3. THE SUBTEST LOOPS CYCLE ON THE PRESENT OLD VALUE  
/BEING INCREMENTED TO THE GOOD VALUE.

1235 0000 TEST04, 0  
1236 4523 CLEAR1

/VERIFY COMPLETE WC INCREMENT USING SPCFWD FUNCTION,  
/START WITH WC=0 AND INCREMENT A FULL COUNT FOUR  
/TIMES DISREGARDING WC OVERFLOW, /SPACE FORWARD;  
T04A, SPCFWD M4

1237 4545 TAD DCA TXXTM1  
1240 1040 DCA GOOD  
1241 3157 DCA GOOD  
1242 3154 DCA GOOD  
1243 1154 TAD OLD  
1244 3156 DCA OLD  
1245 2154 ISZ GOOD  
1246 5252 JMP T04ASL  
1247 2157 ISZ TXXTM1  
1250 7410 SKP  
1251 5262 JMP T4LP6  
1252 1156 TAD OLD  
1253 4473 LWCR  
1254 4535 BREAK  
1255 4503 RWCR  
1256 4533 COMPAR  
1257 7001 ER04A

T04AL, DCA OLD /GOOD TO OLD,  
DCA OLD /+1 TO GOOD,  
T04ASL, TAD OLD /YES,  
LWCR /LOAD WC WITH OLD,  
BREAK /1 BREAK,  
RWCR /CHECK  
COMPAR

/WC INCREMENT FAILURE, GD=GOOD WCJ OD=PREVIOUS GOOD WCJ  
/WC=REAL WC, THE SUBTEST LOOP CYCLES ON THE CURRENT PREVIOUS  
/GOOD WC (OD) BEING INCREMENTED TO THE GOOD WC VALUE,  
JMP T04ASL  
JMP T04AL

1260 5252 T4LP6, LOOP6  
1261 5243 JMP T04A /\*\*\*\*\*LOOP 6\*\*\*\*

/VERIFY THAT "GO" BIT CAN BE SET WITH MITF CLEAR;  
T04B, TAD K100  
DCA GOOD

1264 1027 CLEAR2  
1265 3154 WRITE  
1266 4524 GO  
1267 4543 RFSR K100  
1270 4550 AND  
1271 4507 COMPAR  
1272 0027 ER04B  
1273 4533  
1274 7003

/GO BIT NOT SET, GD=GOOD GO BIT; BD=REAL BIT FROM FS,

1275 5264

JMP

T04B

/SUBTEST LOOP,

/VERIFY COMPLETE CA AND EMA INCREMENT, T04C PERTAINS  
 /TO CA TEST, T04D TO EMA TEST. START WITH  
 /CA=0, EMA=0. CA WILL OVERFLOW, EMA WILL COUNT TO  
 /7 THEN FREEZE. CA SHOULD CONTINUE TO WRAP AROUND,  
 T04CD, TAD M20 /20 PASSES,

1276 1043  
 1277 3157  
 1300 3160  
 1301 3376  
 1302 1160  
 1303 3161  
 1304 1376  
 1305 3377  
 1306 2160  
 1307 5324  
 1310 2157  
 1311 7410  
 1312 5373  
 1313 1376  
 1314 1023  
 1315 0026  
 1316 3376  
 1317 1376  
 1320 7640  
 1321 5324  
 1322 1026  
 1323 3376  
 1324 1160  
 1325 3154  
 1326 4517  
 1327 1161  
 1330 3156  
 1331 1161  
 1332 4475  
 1333 1377  
 1334 4477  
 1335 4456  
 1336 4140  
 1337 4535  
 1340 4504  
 1341 4533  
 1342 7005

DCA TXXTM1  
 DCA TXXTM2  
 DCA T04T1  
 TAD TXXTM2  
 DCA TXXTM3  
 TAD T04T1  
 DCA T04T2  
 ISZ TXXTM2  
 JMP T04CSL  
 ISZ TXXTM1  
 SKP  
 JMP T4LP7  
 TAD T04T1  
 TAD K10  
 AND K70  
 DCA T04T1  
 TAD T04T1  
 SZA CLA  
 JMP I+3  
 TAD K70  
 DCA T04T1  
 TAD TXXTM2  
 GOOD  
 TXXTM3  
 OLD  
 TXXTM3  
 T04T2  
 FUNCP

T04CSL,  
 DCA  
 CLT  
 TAD  
 DCA  
 TAD  
 LCAR  
 TAD  
 LCMR  
 JMS I  
 4140  
 BREAK  
 RCAR  
 COMPAR  
 ER04C

/CLEAR ALL;  
 /SET UP OLD.  
 /LOAD CA WITH OLD VALUE,  
 /LOAD EMA WITH OLD VALUE,  
 /WRITE-GO WITH EMA INC ENABLE,  
 /A BREAK  
 /CHECK CA INCREMENT

/CA INCREMENT FAILURE, CA DID NOT INCREMENT FROM  
 /OLD TO GOOD VALUE. GD=GOOD CAJ OD=OLD CAJ CA=REAL CA;  
 /SUBTEST LOOP CYCLES ON OLD BEING INCREMENTED TO GOOD;

1343 7000  
 1344 7410  
 1345 5357  
 1346 1376  
 1347 3154  
 1350 1377  
 1351 3156  
 1352 4506

T04E  
 T04T1  
 GOOD  
 T04T2  
 OLD  
 RCMR  
 SKP  
 JMP  
 TAD  
 DCA  
 TAD  
 DCA  
 OLD  
 RCMR  
 /PUT GOOD EMA IN GOOD,  
 /OLD EMA IN OLD,  
 /CHECK EMA (CM6-8)

```

1353 4533 COMPAR
1354 7007 ER04D
/EMA INCREMENT OR FREEZE FAILURE, EMA (CM6=0) DID
/NOT INCREMENT FROM OLD TO GOOD VALUE OR DID
/INCREMENT OR CHANGE WHEN IT SHOULD NOT HAVE,
/GD=GOOD EMA; OD=OLD EMA; CM=REAL EMA IN
/BITS 6-8, THE SUBTEST LOOP CYCLES ON CURRENT DATA,
JMP T04CSL /SUBTEST LOOP,
JMP T04CL
1355 5324
1356 5302

```

```

1357 4524 /VERIFY THAT EMA DOES NOT INCREMENT WHEN EMA INC
1360 3154 /DISABLED; CA SET TO 7777, EMA TO 0, THEN CAUSE CA
1361 7240 /INCREMENT AND EMA SHOULD REMAIN 0,
T04E, CLEAR2 /CLEAR ALL,
DCA CMA GOOD /7777 TO CA,
LCAR
LCMR /0 TO EMA
JMS I FUNCPC /WRITE=GO, EMA INC
4100 /DISABLED,
BREAK /1 BREAK,
RCMR /CHECK EMA=0,
COMPAR
ER04E
1370 4533
1371 7011

```

```

1372 5357 /EMA CHANGED WHEN EMA INC DISABLED, GO=GOOD EMA;
/CM=REAL EMA IN BITS 6-8,
JMP T04E /SUBTEST LOOP,
1373 4530
1374 5262 T4LP7, LOOP7
1375 5635 JMP I T4LP6
1376 0000 JMP I TEST04
1377 0000 T04T1, 0
T04T2, 0

```

```

/*****LOOP 7****
/EXIT

```



```

/TEST 05. MEMORY=DATA BUFFER DATA TEST.
/1. VERIFIES THAT DB CAN RECEIVE DATA FROM MEMORY AND
/CAN TRANSFER DATA TO MEMORY.
/2. ALL SUBTEST LOOPS CYCLE ON CONSTANT DATA,
/3. LOOPS 6-7:
/3.1 LOOP 6 CYCLES ON T05A (MEM TO DB=WRITE)
/3.2 LOOP 7 CYCLES ON T05B (DB TO MEM=READ)
/4. LOCATION XBUFF IS LOADED INTO CA
/WHICH CAUSES XBUFF+1 TO BE USED FOR TRANSFERS;
    
```

```

PAGE
TEST05, 0 CLEAR1
1400
0000
1401 4523

1402 4543
1403 4650
1404 7777
1405 5222
1406 1047
1407 4475
1410 1157
1411 3051
1412 1165
1413 4477
1414 4535
1415 4510
1416 4533
1417 7013

1420 5206
1421 5651

1422 4527
1423 5202

1424 4650
1425 7777
1426 5245
1427 4524
1430 1047
1431 4475

/VERIFY PROPER DATA TRANSFERS FROM XBUFF+1 IN MEMORY TO
/DB USING WRITE FUNCTION, CA IS SET TO XBUFF AND
/SHOULD INCREMENT TO XBUFF+1 BEFORE TRANSFER, COMPLEMENT
/INCREMENTING DATA IS USED,
T05A, WRITE /SET WRITE FUNCTION,
JMS I T05LDS /GET DATA,
7777
JMP T5LP6 /DONE EXIT,
T05ASL, TAD XBUFFP /LOAD CA=XBUFF,
LCAR /LOAD XBUFF+1 WITH DATA WORD,
TAD TXXTM1 /LOAD CM WITH PROPER MEMORY
DCA XBUFF+1 /FIELD,
TAD PRGFLD /1 DATA BREAK,
LCMR /READ AND CHECK DB;
BREAK
RBR
COMPAR
ER05A
/GD GOOD DB; 00=PREVIOUS GOOD DB; DB=REAL DB;
JMP T05ASL /SUBTEST LOOP,
JMP I T05LDR

T5LP6, LOOP6 /*****LOOP 6****
JMP T05A

/VERIFY PROPER DATA TRANSFERS FROM DB TO XBUFF+1 (IN MEMORY)
/USING THE READ FUNCTION, CA SET TO XBUFF AND SHOULD
/INCREMENT TO XBUFF+1 BEFORE TRANSFER, COMPLEMENT INCREMENTING
/DATA USED,
T05B, JMS I T05LDS /GET DATA,
7777
JMP T5LP7 /DONE EXIT,
T05BSL, CLEAR2 /CLEAR REGISTERS,
TAD XBUFFP /LOAD CA WITH XBUFF,
LCAR
    
```

1432 1157 TXXTM1 /DATA TO DB,  
 1433 4501 LDBR /FIELD TO CM,  
 1434 1165 TAD /SET HEAD FUNCTION  
 1435 4477 LCMR /1 DATA BREAK,  
 1436 4541 READ /COMPARE MEMORY  
 1437 4535 BREAK  
 1440 1051 TAD XBUFF+1

1441 4533 COMPAR  
 1442 7015 ER05B  
 /GD= GOOD CONTENTS OF XBUFF+1; BDIREAL CONTENTS OF XBUFF+1;  
 /OD= PREVIOUS GOOD CONTENTS OF XBUFF+1;  
 JMP T05BSL /SUBTEST LOOP,  
 JMP I T05LDR

1445 4530 T5LP7, LOOP7 /\*\*\*\*\*LOOP 7\*\*\*\*  
 1446 5222 JMP I T5LP6  
 1447 5600 JMP I TEST05

1450 0707 T05LDS, T02LDS  
 1451 0731 T05LDR, T02LDR

```

/TEST 06, CURRENT ADDRESS MEMORY REFERENCE TEST.
/1, VERIFIES THAT CA AND CM6-8 CAN REFERENCE MEMORY
CORRECTLY IN ALL EXISTING MEMORY FIELDS.
/2, THE METHOD USED INSERTS THE CURRENT ADDRESS AND THEN THE COMPLEMENT
/OF THE ADDRESS IN THE CURRENT MEMORY LOCATION FOR DATA. THE CA AND EMA ARE SET TO THE
/CURRENT LOCATION - 1; THEN A 1 WORD SIMULATED WRITE IS
/EXECUTED; MEMORY IS RESTORED TO ITS ORIGINAL CONTENTS AND THEN
/DB AND DATA ARE COMPARED. DATA IS NOT INSERTED IN LOCATIONS
/MARK1 THROUGH MARK2. THE ACTUAL MEMORY CONTENTS ARE USED FOR DATA.
/3, THE SUBTEST LOOP CYCLES ON THE SAME MEMORY LOCATION
/USING THE SAME DATA.

```

```

PAGE
TEST06, 0 CLEAR1
/VERIFY THAT CA AND EMA (CM6-8) REFERENCE THE CORRECT MEMORY LOCATION
/AND THAT DB RECEIVES CORRECT DATA, WRITE FUNCTION USED, DATA IS ADDRESS
/AND COMPLEMENT ADDRESS.
T06A, JMS T6EX1
      CLT OLD /SET UP CA AND CM.
      TAD LCAR
      LCMR TTXTM2
      JMS I FUNCN /WRITE=GO=EMA INCREMENT ENABLED
      4140 /INTERRUPT SYSTEM TURNED
      IOF /OFF.
      TAD DCA /CHANGE TO CURRENT DATA FIELD.
      DCA /N /GET ORIGINAL MEM AND SAVE.
      TAD I T06AMP
      DCA T6SAVE /GET DATA AND PUT IN MEM.
      TAD TTXTM3
      DCA I T06AMP /DATA BREAK,
      SKP /RESTORE MEMORY.
      6727
      NOP
      TAD DCA I T6SAVE
      DCA I T06AMP /CHANGE TO THIS PROGRAMS DATA FIELD.
      JMS I SCDFP /INTERRUPT SYSTEM ON.
      ION /DATA SHOULD BE IN DB.
      RDBR
      COMPAR
      ER06A
/OD=GOOD DB; OD=CURRENT MEMORY LOCATION BEING USED MINUS ONE; DB=REAL DB
      JMP T06A+1 /SUBTEST LOOP,
      JMS T6EX2
      LOOP6 T06A-1
      JMP /*****LOOP 6****

```

```

1600 4242
1600 4517
1601 1156
1605 4475
1606 1160
1607 4477
1610 4456
1611 4140
1612 6002
1613 1161
1614 3154
1615 6201
1616 1623
1617 3361
1620 1161
1621 3623
1622 7410
1623 0000
1624 6727
1625 7000
1626 1361
1627 3623
1630 4457
1631 6001
1632 4510
1633 4533
1634 7017
1635 5203
1636 4264
1637 4527
1640 5201

```

```

1641 5600 JMP I TEST06
1642 0000 MEMFLD
1643 1167 TAD
1644 3157 DCA
1645 1046 TAD
1646 3215 DCA
1647 3223 T06ADF
1650 1362 DCA MARK1P
1651 7041 CIA
1652 3364 DCA T6CNT1
1653 1364 TAD T6CNT1
1654 1363 TAD MARK2P
1655 7041 CIA
1656 3365 DCA T6CNT2
1657 7040 CMA
1660 3366 DCA T6CNT3
1661 1356 TAD KNOP
1662 3313 DCA T6I1
1663 7410 SKP
1664 0000 T6EX2,
1665 2366 ISE
1666 7610 SKP CLA
1667 5273 JMP CLA
1670 7240 CLA CMA
1671 3367 DCA T6FLAG
1672 5313 JMP T6I1
1673 3367 DCA T6FLAG
1674 1037 TAD M2
1675 3366 DCA T6CNT3
1676 1223 TAD T06AMP
1677 3156 DCA OLD
1700 1215 TAD T06ADF
1701 0026 AND K70
1702 3160 DCA TXXTM2
1703 2223 ISE T06AMP
1704 5313 JMP T6I1
1705 2157 ISE TXXTM1
1706 7410 SKP I
1707 5664 JMP I
1710 1215 TAD T06ADF
1711 1023 TAD K10
1712 3215 DCA T06ADF
1713 0000 T6I1,
/
/
/
1714 1367 SPA CLA
1715 7710 JMP
1716 5347 TAD
1717 1160 CIA
1720 7041 TAD
1721 1165 SNA CLA
?2 7650 PRGFLO
/NO, BYPASS MARK1 UPDATES,
/YES, BYPASS MARK1 UPDATES,
/HAS THE CURRENT ADDRESS REACHED
/THE PROGRAM FIELD?
/NOT UP TO MARK1. TIME FOR DATA COMPLEMENT?
/YES, SET COMPLEMENT DATA
/FLAG, THEN GO TO T6I1
/NO, CLEAR COMPLEMENT FLAG,
/RESET COMPLEMENT COUNTER,
/PUT PREVIOUS ADDRESS IN
/OLD,
/PUT PREVIOUS DATA FIELD
/IN TEMP FOR LOADING
/CM6=0,
/UPDATE TO NEW CURRENT ADDRESS,
/NO OVERFLOW = GO TO T6I1 FOR ROUTING,
/ADDRESS OVERFLOW, CHECK FOR
/ANOTHER EXISTING MEM FIELD,
/NO MORE LEFT, EXIT TO END OF TEST,
/SOME FIELDS LEFT, UPDATE
/OPF INSTRUCTION FOR NEW
/FIELD,
/THIS INSTRUCTION CAN BE ONE OF THREE INSTRUCTIONS:
/PRIOR TO REACHING MARK1;
/AFTER REACHING MARK1 BUT NOT MARK2;
/AFTER PASSING MARK2,
/NOT UP TO MARK1. TIME FOR DATA COMPLEMENT?
/YES, SET COMPLEMENT DATA
/FLAG, THEN GO TO T6I1
/NO, CLEAR COMPLEMENT FLAG,
/RESET COMPLEMENT COUNTER,
/PUT PREVIOUS ADDRESS IN
/OLD,
/PUT PREVIOUS DATA FIELD
/IN TEMP FOR LOADING
/CM6=0,
/UPDATE TO NEW CURRENT ADDRESS,
/NO OVERFLOW = GO TO T6I1 FOR ROUTING,
/ADDRESS OVERFLOW, CHECK FOR
/ANOTHER EXISTING MEM FIELD,
/NO MORE LEFT, EXIT TO END OF TEST,
/SOME FIELDS LEFT, UPDATE
/OPF INSTRUCTION FOR NEW
/FIELD,
/THIS INSTRUCTION CAN BE ONE OF THREE INSTRUCTIONS:
/PRIOR TO REACHING MARK1;
/AFTER REACHING MARK1 BUT NOT MARK2;
/AFTER PASSING MARK2,
/NOT UP TO MARK1. TIME FOR DATA COMPLEMENT?
/YES, BYPASS MARK1 UPDATES,
/HAS THE CURRENT ADDRESS REACHED
/THE PROGRAM FIELD?
/NO, BYPASS MARK1 UPDATES,

```

/TEST 06 LOCAL EXECUTIVE;  
/GET EXTENDED MEM FIELDS  
/AND PUT IN TEMP.  
/INITIALIZE OPF INSTRUCTION,  
/SET STARTING ADDRESS TO 0,  
/MAKE THE ADDRESS MARK1 INTO  
/A COUNTER,  
/MAKE (MARK2-MARK1)+2 INTO  
/A COUNTER,  
/FAKE THE COMPLEMENT COUNTER SO  
/IT RESETS ON FIRST ENTRY,  
/SET T6I1 TO NOP,  
/ENTER HERE AT END OF EACH PASS,  
/TIME FOR COMPLEMENT DATA?  
/YES, SET COMPLEMENT DATA  
/FLAG, THEN GO TO T6I1  
/NO, CLEAR COMPLEMENT FLAG,  
/RESET COMPLEMENT COUNTER,  
/PUT PREVIOUS ADDRESS IN  
/OLD,  
/PUT PREVIOUS DATA FIELD  
/IN TEMP FOR LOADING  
/CM6=0,  
/UPDATE TO NEW CURRENT ADDRESS,  
/NO OVERFLOW = GO TO T6I1 FOR ROUTING,  
/ADDRESS OVERFLOW, CHECK FOR  
/ANOTHER EXISTING MEM FIELD,  
/NO MORE LEFT, EXIT TO END OF TEST,  
/SOME FIELDS LEFT, UPDATE  
/OPF INSTRUCTION FOR NEW  
/FIELD,  
/THIS INSTRUCTION CAN BE ONE OF THREE INSTRUCTIONS:  
/PRIOR TO REACHING MARK1;  
/AFTER REACHING MARK1 BUT NOT MARK2;  
/AFTER PASSING MARK2,  
/NOT UP TO MARK1. TIME FOR DATA COMPLEMENT?  
/YES, BYPASS MARK1 UPDATES,  
/HAS THE CURRENT ADDRESS REACHED  
/THE PROGRAM FIELD?  
/NO, BYPASS MARK1 UPDATES,

```

1723 2364      ISZ      T6CNT1
1724 5347      JMP      T6EX4
1725 1357      TAD      KJMP61
1726 3313      DCA      T6I1
1727 1367      TAD      T6FLAG
T6EX3:
1730 7710      SPA      CLA
1731 5337      JMP      T6EX5
1732 2365      ISZ      T6CNT2
1733 5337      JMP      T6EX5
1734 1360      TAD      KJMP62
1735 3313      DCA      T6I1
1736 5347      JMP      T6EX4
T6EX5:
1737 1215      TAD      T06ADF
1740 3342      DCA      .+2
1741 6002      IOF
1742 6201      CDF
1743 1623      TAD      I
1744 4457      JMS      I
1745 6001      ION
1746 5353      JMP
T6EX4:
1747 1223      TAD
1750 2367      ISZ
1751 7040      CMA
1752 7040      CMA
T6EX6:
1753 3161      DCA
1754 4474      CHCR
1755 5642      JMP      I

KNOP,      NOP
KJMP61,    JMP
KJMP62,    JMP
T6SAVE,    0
MARK1P,    MARK1
MARK2P,    MARK2+2

T6CNT1,    0
T6CNT2,    0
T6CNT3,    0
T6FLAG,    0

/NO, UPDATE MARK1 COUNTER,
/NOT THERE YET, USE DATA,
/AT MARK1, CHANGE ROUTING TO T6EX3;
/PASSED OR AT MARK1, COMPLEMENT
/DATA PASS;
/YES, BYPASS (MARK2=MARK1)+2 UPDATES,
/NO, ARE WE PASSED MARK2?
/NO, USE ACTUAL MEM CONTENTS,
/YES, CHANGE ROUTING TO EX4,

/USE MEM CONTENTS VICE DATA, GET CURRENT
/DOF INSTRUCTION AND STORE FOR USE;
/INTERRUPT SYSTEM OFF,
/CHANGE TO CURRENT OF;
/GET CONTENTS OF CURRENT ADDRESS,
/RESET TO OF OF PROGRAM,
/INTERRUPT SYSTEM ON,
/PUT MEM CONTENTS IN TEMP AND EXIT;
/USE DATA, GET CURRENT ADDRESS
/FOR DATA, TIME FOR COMPLEMENT?
/NO, COMPLEMENT DATA ONCE,
/YES, DITTO OR COMPLEMENT DATA AGAIN,
/STORE DATA OR MEM CONTENTS IN TEMP,

/USED FOR STORING DATA IN CURRENT
/LOCATION, THEN RUN TEST,

/BIASED BY +2 TO YIELD CORRECT TALLY
/NUMBER FOR (MARK2=MARK1)+2,
/MARK1 COUNTER,
/(MARK2=MARK1)+2 COUNTER,
/COMPLEMENT DATA COUNTER,
/SET TO 7777 IF TIME FOR COMPLEMENT DATA.

```

/TEST 07: STATUS REGISTERS STAND ALONE TEST.

/1. THE PURPOSE OF THIS TEST IS TO VERIFY THAT MS AND FS INDICATE ONLY WHAT IS EXPECTED WITH THE TRANSPORT SYSTEM EFFECTIVELY DISCONNECTED FROM THE TMBE CONTROL. ALL INDICATORS WHICH MAY BE IN ONE STATE OR ANOTHER ARE DISREGARDED.

/2. LOOPS 6-7:

/2.1 LOOP 6 CYCLES ON MS CHECK (T07A=T07B)  
/2.2 LOOP 7 CYCLES ON S1 CHECK (T07C=T07D)

2000	PAGE	
2000	TEST07, 0	
2001	CLEAR1	
2002	T07A,	/SELECT REMOTE BIT TO GOOD;
2003	TAD	K400
2004	DCA	GOOD
2005	RMSR	
2006	AND	K3774
2007	COMPAR	
	ER07A	
2010	/GD=GOOD MS MASKED; BD=REAL MS MASKED;	
	JMP	T07A+2
2011	/VERIFY THAT RMSR CLEARS AC BEFORE READING; SELECT REMOTE	
2012	/SHOULD BE ONLY BIT SET.	
2013	T07B,	
2014	CLA CMA	
2015	RMSR	
2016	AND	K3774
2017	COMPAR	
2018	ER07B	
2019	/GD=GOOD MS MASKED; BD=REAL MS MASKED;	
2020	JMP	T07B
2021	LOOP6	
2022	JMP	T07A
2023		/*****LOOP6****
2024	/FS SHOULD INDICATE BIT 6 SET SINCE NO DRIVE ON LINE	
2025	/DEFAULTS TO 9 TRK.	
2026	T07C,	
2027	OFFLIN	
2028	TAD	K40
2029	DCA	GOOD
2030	TAD	K3
2031	LCMR	
2032	RFSR	
2033	COMPAR	
2034	ER07C	
2035	/GD=GOOD FS; BD=REAL FS;	
2036	JMP	T07C+3
2037	/VERIFY THAT RFSR CLEARS AC PRIOR TO READING.	
2038		/SUBTEST LOOP.
2039		
2040		/LOAD 9 TRK 800BPI
2041		/FUNCTION TO 00.

```

2032 7240 T070, CLA CMA /AC=7777,
2033 4507 RFSR /READ FS-9TRK BIT SHOULD
2034 4533 COMPAR /STILL BE SET,
2035 7027 ER07D
/GD=GOOD FS; BD=REAL FS. /SUBTEST LOOP
2036 5232 JMP T07D

/VERIFY THAT CM BIT 11 = 0 FORCES 7 CHANNEL. (FS BIT 6 = 0)
2037 3154 DCA GOOD /DENSITY BITS=10
2040 7126 SIL RTL
2041 4477 LCMR
2042 4507 RFSR
2043 4533 COMPAR
2044 7031 ER07E
/GD=GOOD FS; BD=REAL FS. /SUBTEST LOOP,
2045 5240 JMP T07E+1

LOOP7 /*****LOOP 7****
2046 4530 JMP T07C-2
2047 5217

K3774, JMP I TEST07
2050 5600
2051 3774

```

/TEST 10, BASIC TESTS FOR MTF, GO BIT, IF, R/C ERROR, EMA OVERFLOW  
 /AND EF, /VERIFIES EXISTENCE AND PROPER FUNCTIONING OF THE ABOVE ON A  
 /PARTIAL BASIS,

/LOOPS 6-9;  
 /LOOP 6 CYCLES ON MTF AND GO BIT TESTS (T10A-T10J)  
 /LOOP 7 CYCLES ON IF TESTS (T10K-T10R)  
 /LOOP 8 CYCLES ON R/C AND EMA CHECKS (T10S-T10W)  
 /LOOP 9 CYCLES ON EF CHECKS (T10X-T10Z)

2200	PAGE		
2201	TEST10, 0		
	CLEAR1		
2202	JMS	T10A	/EXECUTE T10A
2203	JMP	T10B	/THEN GO TO T10B
2204	0		/VERIFY CLI CLEARS MTF, CHECK VIA SKTD NOT SKIPPING,
2205	DCA	GOOD	/JMS'D HERE
2206	CLEAR2		/CLI
2207	SKTD		/SHOULD NOT SKIP.
2210	SKP		
2211	CMA		
2212	COMPAR		
2213	ER10A	T10A+2	/SUBTEST LOOP.
2214	JMP	T10A	
2215	JMP I		
2216	2216		/VERIFY WCOV OCCURS WITH WC=7777, SPACE FORWARD THEN
2217	4553		/1 DATA BREAK FOLLOWED BY LOBR, LOBR,WCOV SHOULD SET MTF AND
2220	4501		/SKTD SHOULD SKIP,
2221	4514		T10B, SPCFWD
2222	7040		/SPACE FORWARD,
2223	4533		/FORCE WCOV
2224	7035		/SET MTF
2225	5216		/MTTF SHOULD BE SET CAUSING
			/SKTD TO SKIP,
			/SUBTEST LOOP,
2226	4204	JMS	/T10A AGAIN
2227	7240	CLA	/SET UP FOR NEXT.
2230	3154	DCA	GOOD
2231	4524		/VERIFY LOBR WITH NO WCOV DOES NOT SET MTF,
2232	4473		/CHECK VIA SKTD,
33	4545		T10C, CLEAR2
34	4550		LWCR
			SPCFWD
			GO
			/CLEAR ALL,
			/ENSURE WC=0000
			/SPACE FORWARD FUNCTION,
			/GO.



2235 4535 BREAK  
 2236 4501 LDBR  
 2237 4514 SKTD  
 2240 7040 CMA  
 2241 4533 COMPAR  
 2242 7037 ER10C  
 2243 5231 JMP T10C  
 /SUBTEST LOOP,  
 /SKTD SHOULD NOT SKIP  
 /1 DATA BREAK,

2244 4524  
 2245 4761  
 2246 7477  
 2247 7753  
 2250 2253  
 2251 4545  
 2252 4553  
 2253 0000  
 /VERIFY WCOV AND AN IOT OTHER THAN LDBR  
 /DOES NOT SET MTF, VARIABLE IOT (I1) INSERTED BY EXECUTIVE,  
 T10E,  
 CLEAR2  
 JMS I T10X1P /GET IOT,  
 T10LS1-1  
 -25  
 .43  
 SPCFWD  
 WCOV  
 0  
 /PUT IT HERE;

2254 7000 NOP  
 2255 7200 CLA  
 2256 4514 SKTD  
 2257 7040 CMA  
 2260 4533 COMPAR  
 2261 7041 ER10D  
 /I1=VARIABLE IOT  
 JMP I T10X2P /SUBTEST LOOP,  
 JMS I T10X3P /GET MORE IOT'S,  
 /SKTD SHOULD NOT SKIP,

2264 4524  
 2265 4540  
 2266 4514  
 2267 7040  
 2270 4533  
 2271 7043  
 2272 5264  
 /VERIFY THAT REWIND FUNCTION WITHOUT RW STATUS (SINCE  
 /NO DRIVE ON LINE) DOES NOT SET MTF,  
 T10E,  
 CLEAR2  
 REWIND  
 SKTD  
 CMA  
 COMPAR  
 ER10E  
 JMP T10E  
 /SHOULD NOT SKIP,  
 /SUBTEST LOOP,

2273 4524  
 2274 4546  
 2275 4514  
 2276 7040  
 2277 4533  
 2300 7045  
 2301 5273  
 /VERIFY SPACE REVERSE WITHOUT BUT DOES NOT SET MTF,  
 T10F,  
 CLEAR2  
 SPCREV  
 SKTD  
 CMA  
 COMPAR  
 ER10F  
 JMP T10F  
 /SUBTEST LOOP,

2302 4524  
 2303 4541  
 2304 4542  
 /VERIFY READ, READ COMPARE OR WRITE WITHOUT LPCS DOES NOT SET  
 /MTF,  
 T10G,  
 CLEAR2  
 READ  
 RDCOMP

```

2305 4543 WRITE
2306 4514 SKTD
2307 7040 CMA /SHOULD NOT SKIP.
2310 4533 COMPAR
2311 7047 ER10G
2312 5302 JMP T10G /SUBTEST LOOP,

```

```

2313 1027 /VERIFY "GO" BIT SETS WHEN LOADED AND MTF CLEAR,
2314 3154 T10H, TAD K100
2315 4524 DCA GOOD
2316 4550 CLEAR2
2317 4507 GO
2320 0027 RFSR K100
2321 4533 COMPAR
2322 7051 ER10H
2323 5315 JMP /GD= GOOD "GO" BIT; BD= REAL "GO" BIT, /SUBTEST LOOP,

```

```

2324 3154 /VERIFY "GO" BIT CLEARS AFTER LOADED WITH MTF SET;
2325 4545 T10I, DCA GOOD
2326 4553 SPCFWD /SET MTF,
2327 4501 WCOV /SET MTF;
2330 4550 LDBR /SET "GO"/ SHOULD CLEAR
2331 4507 GO /IMMEDIATELY,
2332 0027 AND K100
2333 4533 COMPAR
2334 7053 ER10I
2335 5325 JMP /GD= GOOD "GO" BIT; BD= REAL GO BIT, /SUBTEST LOOP,

```

```

2336 4524 /VERIFY CLT CLEARS "GO" BIT.
2337 4550 T10J, CLEAR2
2340 4524 GO /SET "GO"
2341 4507 CLEAR2 /CLT
2342 0027 RFSR /CHECK CLEAR;
2343 4533 AND K100
2344 7055 COMPAR
2345 5337 ER10J
JMP /GD= GOOD "GO" BIT; BD= REAL "GO" BIT, /SUBTEST LOOP,

```

```

2346 4527 LOOP6
2347 5201 JMP TEST10+1 /*****LOOP 6****

```

```

2350 3154 /VERIFY CLT CLEARS ILLEGAL FUNCTION (IF), ACTUALLY IT SHOULD
2351 4524 /NEVER BE SET IN THIS TEST - SEE T100.
2352 4505 T10K, DCA GOOD
2353 0020 RMSR
2354 4533 AND K1
2355 4533 COMPAR

```

```

2355 7057 ER10K
/GD= GOOD "IF"; BD= REAL "IF"
JMP T10K+1 /SUBTEST LOOP

```

```

2357 5760 JMP I ,*1
2360 2400 T100
2361 1213 T10X1P; T3EX1
2362 1227 T10X2P; T3EX3+2
2363 1231 T10X3P; T3EX2

```

2400 PAGE

/VERIFY WRITE OR WEOF WITHOUT FILE PROTECT DOES NOT SET IF.

```

2400 4524 T100, CLEAR2
2401 4543 WRITE
2402 4544 WEOF
2403 4505 RMSR
2404 0020 AND K1
2405 4533 COMPAR
2406 7061 ER100
/GD= GOOD IF; BD= REAL IF (BIT 11)
JMP T100 /SUBTEST LOOP

```

/VERIFY LCMR, LFGR OR LOBR WITHOUT CONTROL BUSY DO NOT

```

2410 4524 T10P, CLEAR2
2411 4477 LCMR
2412 4500 LFGR
2413 4501 LOBR
2414 4505 RMSR
2415 0020 AND K1
2416 4533 COMPAR
2417 7063 ER10P
/GD= GOOD IF; BD= REAL IF (BIT 11)
JMP T10P /SUBTEST LOOP

```

/VERIFY 9 CHANNEL AND CM10=0 AND "GO" DO NOT SET IF.

/("GO" SHOULD NOT GENERATE "PRESEI" NOW.)

```

2421 4524 T100, CLEAR2
2422 4550 GO /CM10=0 FROM CLEAR;
2423 4505 RMSR
2424 0020 AND K1
2425 4533 COMPAR
2426 7065 ER100
2427 5221 JMP T100 /SUBTEST LOOP.

```

/VERIFY SPCREV WITHOUT BOT DOES NOT SET IF.

```

2430 4524 T10R, CLEAR2
2431 4546 SPCREV
2432 4505 RMSR

```

```

2433 0020      AND      K1
2434 4533      COMPAR
2435 7067      ER10R
          /GD= GOOD IF; BD= REAL IF (BIT 11)
2436 5230      JMP      T10R      /SUBTEST LOOP,

2437 4530      LOOP7
2440 5752      JMP I   T10KP      /*****LOOP 7*****

2441 4243      JMS     T10S      /EXECUTE T10S
2442 5254      JMP     T10T      /THEN ONTO T10T,

2443 0000      /VERIFY "CLT" CLEARS READ/COMPARE ERROR,
2444 3154      DCA     GOOD
2445 4524      CLEAR2
2446 4505      RMSR
2447 0021      AND     K2
2448 4533      COMPAR
2449 7071      ER10S

2452 5245      /GD= GOOD R/C ERROR BIT; BD= REAL R/C ERROR (BIT 10)
2453 5643      JMP     T10S+2
          JMP I   T10S

2454 1021      /VERIFY THAT DB=7777 AND MEM=0000 YIELDS AN R/C ERROR,
2455 3154      TAD     K2
2456 3051      DCA     GOOD
          DCA     XBUFFER+1      /MEM IO 0000
2457 4524      CLEAR2
          TAD     XBUFFER      /MEM+1 TO CA
2460 1047      LQAR
2461 4475      TAD     PRGFLO      /FIELD TO CM,
2462 1165      TAD     CMA      /7777 IO DB
2463 4477      LDBR
2464 7040      RDCOMP
2465 4501      BREAK
2466 4542      RMSR
2467 4535      AND     K2
2470 4505      COMPAR
2471 0021      ER10T
2472 4533      /GD= GOOD R/C ERROR; BD= REAL R/C ERROR (BIT 10)
2473 7073      JMP     T10T+3      /SUBTEST LOOP

2474 5257      JMS     T10S
2475 4243      JMS     T10U
2476 4300      JMP     T10V

2500 0000      /VERIFY CLT CLEARS EMA OVERFLOW BIT,
2501 3154      T10U,
2502 4524      DCA     GOOD
          CLEAR2
          RFSR
          3 4507
    
```

```

2504 0023 AND K10
2505 4533 COMPAR
2506 7075 ER10U
/GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
      T10U+2
      JMP I T10U
2507 5302
2510 5700

2511 1023
2512 3154
2513 4524
2514 7040
2515 4475
2516 4536
2517 0070
2520 4456
2521 0140
2522 4476
2523 4507
2524 0023
2525 4533
2526 7077

2527 5313
2530 4300

2531 4524
2532 4476
2533 4536
2534 0070
2535 4456
2536 0140
2537 7040
2540 4475
2541 4507
2542 0023
2543 4533
2544 7101
2545 5331
2546 4531
2547 5237

2550 5751
2551 2600
2552 2346

2600 2600
2602 4202

```

```

/VERIFY EMA OVERFLOW BIT SETS WHEN EMA SET TO 7
/AND CA 0 CHANGES FROM 1 TO 0,
T10V,
DCA K10
GOOD
CLEAR2
CMA
LCAR
COMAND
70
JMS I FUNC
0140
CCAR
RFSR
AND K10
AND COMPAR
ER10V
JMP
/GD= GOOD EMA OVERFLOW; BD= REAL EMA OVERFLOW (BIT 8)
      T10V+2
      JMP
JMS T10U
/VERIFY EMA SET TO 7 AND CA 0 CHANGING FROM 0 TO 1
/DOES NOT SET EMA OVERFLOW BIT,
T10W,
CLEAR2
CCAR
COMAND
70
JMS I FUNC
0140
CMA
LCAR
RFSR
AND K10
AND COMPAR
ER10W
JMP
/GD= GOOD EMA OVERFLOW (BIT 8)
      T10W
      JMP
T10LP8, LOOP8
T10S-4
JMP I 1
T10X-2
T10KP, T10K-2
PAGE
JMS T10X

```

```

/7777 TO CA
/7 TO EMA
/"GO" AND EMA INCREMENT
/ENABLE TO FUNC.
/0 TO CA.
/CHECK
/0 TO CA.
/7 TO EMA
/"GO" AND EMA INCREMENT
/ENABLE TO FUNC.
/7777 TO CA.
/CHECK

```

```

/*****LOOP 8****

```

2601 5213

JMP T10Z

/VERIFY CLI CLEARS EF,  
T10X,

2602 0000  
2603 3154  
2604 4524  
2605 4505  
2606 0034  
2607 4533  
2610 7103

DCA GOOD  
CLEAR2  
RMSR  
AND K4000  
COMPAR  
ER10X

/GD= GOOD EF; BD= REAL EF (BIT 0)  
JMP T10X+2 /SUBTEST LOOP,  
JMP I T10X

2611 5204  
2612 5602

/VERIFY R/C ERROR WITH MTF SET SETS EF,  
T10Z,

2613 1034  
2614 3154  
2615 4524  
2616 1047  
2617 4475  
2620 1165  
2621 4477  
2622 7040  
2623 4501  
2624 4542  
2625 4535  
2626 4545  
2627 4553  
2630 4501  
2631 4505  
2632 0034  
2633 4533  
2634 7105

TAD K4000  
DCA GOOD  
CLEAR2  
TAD XBUFFP /GENERATE R/C ERROR,  
LCAR PRGFLO  
LCMR  
CMA  
LDBR  
RDCOMP  
BREAK  
SPCFWD  
WCOV  
LDBR  
RMSR  
AND K4000  
COMPAR  
ER10Z

/GD=GOOD EF; BD=REAL EF(BIT 0)  
JMP T10Z+2 /SUBTEST LOOP  
JMS T10X

2635 5215  
2636 4202

/VERIFY EMA OVERFLOW AND MTF SET SETS EF,  
T101,

2637 1034  
2640 3154  
2641 4524  
2642 7240  
2643 4475  
2644 4536  
2645 0070  
2646 4456  
2647 0140  
2650 4476  
2651 4545  
2652 4553  
2653 4505  
2654 0034  
5 4533

TAD K4000  
DCA GOOD  
CLEAR2  
CLA CMA  
LCAR  
COMAND  
70  
JMS I FUNC  
0140  
CCAR  
SPCFWD  
WCOV  
RMSR  
AND K4000  
COMPAR  
/FORCE MTF  
/CHECK  
/GENERATE EMA OVERFLOW

```

2656 7107 ER101
2657 5241 /GD=GOOD EF; BD=REAL EF (BIT 0)
          JMP T101+2 /SUBTEST LOOP

2660 3154 /VERIFY EMA OVERFLOW WITHOUT MTF SET DOES NOT SET EF,
2661 4524 T102, DCA GOOD
2662 7240 CLEAR2
2663 4475 CLA CMA
2664 4536 LCAR
2665 0070 COMMAND
2666 4456 70 JMS I FUNCP

2667 0140 0140
2670 4476 CCAR
2671 4505 RMSR
2672 0034 AND K4000
2673 4533 COMPAR
2674 7111 ER102
          /GD=GOOD EF; BD=REAL EF (BIT 0)
          JMP T102+1 /SUBTEST LOOP,

2675 5261

2676 4532 LOOP9
2677 5705 JMP I T10LPP /*****LOOP9*****/

2700 1704 TAD I TST10P
2701 3303 DCA .+2
2702 5703 JMP I .+1
2703 0000 0
2704 2200 TST10P, TEST10
2705 2546 T10LPP, T10LP8

```

/TESI11, MTF AND EF SKIP=NO SKIP TEST,  
 /1, VERIFIES PROPER SKIP OR NO SKIP OF ALL 672X IOT'S  
 /WITH MTF AND EF BOTH CLEAR, THEN EACH SET WITH THE OTHER CLEAR.

/2, ALL IOT'S USED (672X) ARE SUPPLIED BY A LOCAL  
 /EXECUTIVE AND IF AN ERROR OCCURS THE IOT IN USE IS DISPLAYED  
 /UNDER"!!" IN THE ERROR PRINTOUT,

/3, ALL SUBTEST LOOPS CYCLE ON IOT IN PRESENT USE;

/4, LOOPS 6-8:  
 /4.1 LOOP 6 CYCLES ON T11A-T11B (MTF AND EF CLEAR)  
 /4.2 LOOP 7 CYCLES ON T11C-T11D (MTF SET, EF CLEAR)  
 /4.3 LOOP 8 CYCLES ON T11E-T11F (MTF SET, EF SET)

PAGE	TEST	DESCRIPTION
3000	TEST11, 0	GOOD
3000	0	CLEAR1
3001	4523	DCA
3002	3154	GOOD
3003	4524	
3004	4513	
3005	7040	
3006	4533	
3007	7113	
3010	5203	
3011	4322	
3012	3147	
3013	7772	
3014	3016	
3015	4524	
3016	0000	
3017	7410	
3020	7040	
3021	4533	
3022	7115	
3023	5215	
3024	4341	
3025	4527	
3026	5203	
3027	4322	
3030	3146	
3031	7776	
3032	3037	
3011	4322	
3012	3147	
3013	7772	
3014	3016	
3015	4524	
3016	0000	
3017	7410	
3020	7040	
3021	4533	
3022	7115	
3023	5215	
3024	4341	
3025	4527	
3026	5203	
3027	4322	
3030	3146	
3031	7776	
3032	3037	



```

3033 4524 CLEAR2 /FORCE MTF
3034 4545 SPCFWD
3035 4553 WCOV
3036 4501 LDBR /SET MTF,
3037 0000 /IOT SHOULD SKIP,
3040 7040 CMA
3041 4533 COMPAR
3042 7117 ER11C
/11=FAILING IOT, T11C+4
JMP T11EX2
JMS
3043 5233
3044 4341

```

```

3045 4322 /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN
3046 3151 /MTTF SET AND EF SET,
3047 7774 T11D, T11LS1+3 /GET AND IOT
3050 3055 -4
T11D1 /FORCE MTF,
3051 4524 CLEAR2 /SET MTF (ALSO SETS IF AND EF)
3052 4545 SPCFWD /IOT SHOULD NOT SKIP,
3053 4553 WCOV
3054 4501 LDBR
3055 0000
3056 7410 SKP
3057 7040 CMA
3060 4533 COMPAR
3061 7121 ER11D
/11=FAILING IOT
JMP T11D+4 /SUBTEST LOOP,
JMS T11EX2
LOOP7 /*****LOOP7****
JMP T11C-2

```

```

3066 4322 /VERIFY SKEF,SKTD AND SKCB SKIP WHEN MTF AND EF SET,
3067 3145 T11E, JMS T11EX1
3070 7775 T11LS1=1
3071 3074 -3
T11E1
3072 4524 CLEAR2 SETEF
3073 4357 JMS
3074 0000 T11E1,
3075 7040 CMA
3076 4533 COMPAR
3077 7123 ER11E
/11=FAILING IOT, T11E+4
JMP T11EX2
JMS
3100 5272
3101 4341

```

```

3102 4322 /VERIFY SKTR,SDLE,SBRM AND CLF DO NOT SKIP WHEN MTF
3103 3151 /AND EF SET,
T11F, JMS T11EX1
T11LS1+3 /GET AN IOT,

```



3165 1165  
 3166 4477  
 3167 4542  
 3170 4535  
 3171 4545  
 3172 4553  
 3173 4501  
 3174 5757

TAD  
 LCMR  
 RDCOMP  
 BREAK  
 SPCFWD  
 WCOV  
 LDBR  
 JMP I  
 PRGFLD  
 SETEF

/TEST 12, MTTF=EF INTERRUPT TEST,

/1, VERIFIES CORRECT INTERRUPT NON-OCCURRENCES AND OCCURRENCES  
/WITH ALL INTERESTING COMBINATIONS OF MTTF, EF AND THEIR ENABLES,

/2, IF AN ERROR OCCURS, THE CM AND MS PRINTOUTS SHOW  
/THE CURRENT COMBINATION,

/3, ALL SUBTEST LOOPS CYCLE ON CURRENT COMBINATIONS,

/4, LOOPS 6-8:

/4.1 LOOP 6 CYCLES ON T12A (MTTF AND EF CLEAR)  
/4.2 LOOP 7 CYCLES ON T12B-T12C (MTTF SET, EF CLEAR)  
/4.3 LOOP 8 CYCLES ON T12D-T12E (MTTF SET, EF SET,)

3200	PAGE	
3200	TEST12, 0	
3201	CLEAR1	
3202	IOF	
3203	JMS I INSRSP	
3204	T12INT	
3205	T12INT	
3206	DCA	GOOD

/VERIFY NO INTERRUPT WHEN EF AND MTTF CLEAR REGARDLESS OF STATE OF

/INTERRUPT ENABLE BITS,

3207	T12A,	JMS	/GET ENABLES;
3210		T12EX1	
3211		T12LS1=1	
3212		-3	
3213		T12A1	
3214		CLEAR3	/LOAD CM WITH ENABLES,
3215		COMAND	
3216	T12A1,	0	
3217		JMS	/CHECK FOR NO INTERRUPT,
3220		COMPAR	
		ER12A	
3221		JMS	/SUBTEST LOOP,
3222		T12A+4	
		T12EX2	
3223		LOOP6	
3224		JMP	/*****LOOP 6****
		T12A-1	

/VERIFY NO INTERRUPT WHEN MTTF AND EF SET, AND BOTH INTERRUPT ENABLES CLEAR,

T12B,

3225	DCA	GOOD	
3226	CLEAR3		/CLEAR ALL;
3227	SPCFWD		/FORCE MTTF
3230	WCOV		
3231	LDBR		/CLEAR ENABLES
3232	COMAND		
3233	0		/CHECK
3234	JMS	T12EX4	TERRUPT

```

3235 4533 COMPAR
3236 7131 ER12B
3237 5226 JMP T12B+1 /SUBTEST LOOP

/VERIFY INTERRUPT WITH MTF SET AND MITF INT ENABLE SET.
T12C,
3240 7240 CLA CMA
3241 3154 DCA GOOD
3242 4525 CLEAR3
3243 4545 SPCFWD
3244 4553 WCOV
3245 4501 LDBR
3246 4536 COMAND
3247 0100 100
3250 4326 JMS
3251 4533 COMPAR
3252 7133 ER12C
3253 5242 JMP T12C+2 /SUBTEST LOOP

LOOP7
3254 4530 JMP T12B-2 /*****LOOP 7****
3255 5223

```

```

/VERIFY NO INTERRUPT WITH EF AND MTF SET; AND ENABLES CLEAR.
T12D,
3256 3154 DCA
3257 4525 CLEAR3
3260 4745 JMS I SETEFP
3261 4536 COMAND
3262 0000 0
3263 4326 JMS
3264 4533 COMPAR
3265 7135 ER12D
3266 5257 JMP T12D+1 /SUBTEST LOOP

/VERIFY NO INTERRUPT WITH EF AND MTF SET; AND ENABLES CLEAR.
T12E,
3267 7240 CLA CMA
3270 3154 DCA GOOD
3271 4525 CLEAR3
3272 4745 JMS I SETEFP
3273 4536 COMAND
3274 0200 200
3275 4326 JMS
3276 4533 COMPAR
3277 7137 ER12E
3300 5271 JMP T12E+2 /SUBTEST LOOP

LOOP8
3301 4531 JMP T12D-2 /*****LOOP 8****
3302 5254

```

```

/VERIFY INTERRUPT WHEN EF AND EF INT ENABLE SET.
T12E,
3303 4523 CLEAR1
3304 5600 JMP I TEST12

/EXIT
3305 0000 T12EX1, 0 /TEST 12 COMMAND REGISTER CONSTANT
3306 1705 TAD I T12EX1 /SELECTOR, SAME BASIC METHOD
3307 3010 DCA 10

```

3310	2305	ISZ	T12EX1	
3311	1705	TAD I	T12EX1	
3312	3157	DCA	TXXTM1	
3313	2305	ISZ	T12EX1	
3314	1705	TAD I	T12EX1	
3315	3160	DCA	TXXTM2	
3316	2305	ISZ	T12EX1	
3317	1410	T12EX3, TAD I	10	
3320	3560	DCA I	TXXTM2	
3321	5705	JMP I	T12EX1	
3322	0000	0		
3323	2157	ISZ	TXXTM1	
3324	5317	JMP	T12EX3	
3325	5722	JMP I	T12EX2	
3326	0000	0		
3327	7200	CLA		
3330	6001	ION		
3331	7000	NOP		
3332	7000	NOP		
3333	6002	IOF		
3334	5726	JMP I	T12EX4	
3335	7240	CLA CMA		
3336	5726	JMP I	T12EX4	
3337	0000	0		
3340	0100	100		
3341	0200	200		
3342	0000	0		
3343	0200	200		
3344	0100	100		
3345	3157	SETEFP, SETEF		

```

/PUT 0'S IN INTERRUPT INDICATOR.
/(THE AC, )
/INTERRUPT ON.
/WAIT.
/INTERRUPT OFF, AND
/GET INDICATOR AND
/EXIT.
/MTTF, EF INT, HANDLER FOR TEST 12.
/JUST SET AC=7777.
/NO INT ENABLE.
/MTTF INT ENABLE.
/EF INT ENABLE.
    
```

```

/TEST 13, 672X IOT UNIQUENESS FOR CLF AND SBRM,
/1, VERIFIES THAT CLF CLEARS ONLY STATUS BITS AND NOT NON-
/STATUS REGISTERS (SINCE TUR HIGH) AND THAT NO OTHER 672X IOT
/GENERATES CLF OR SBRM,
/2, PORTIONS OF THIS TEST ARE RUN BY TEST 11 LOCAL EXECUTIVE
/IN WHICH CASE THE IOT IN USE IS DISPLAYED UNDER "I1:" IF
/AN ERROR OCCURS,
/3, LOOPS 6-7:
/3.1 LOOP 6 CYCLES ON T13A-T13C (CLF CHECKS)
/3.2 LOOP 7 CYCLES ON T13D-T13E (SBRM UNIQUENESS)

```

```

3400 PAGE
3400 TEST13, 0 CLEAR1
3401 4523

3402 3154
3403 4545
3404 4553
3405 4501
3406 4516
3407 4514
3410 7410
3411 7040
3412 4533
3413 7141
3414 5203

/USING MTF, VERIFY CLF CLEARS STATUS, MTF AND EF; THIS
/IS ENOUGH TO VERIFY THAT CLF IS WORKING.
T13A, DCA GOOD /FORCE MTF
      SPCFWD
      WCOV
      LOBR
      CLF /SHOULD CLEAR MTF
      SKTD /CHECK
      SKP /SUBTEST LOOP,
      CMA
      COMPAR
      ER13A
      JMP T13A+1

/VERIFY IOT 672X EXCEPT CLF DOES NOT CLEAR MTF.
T13B, JMS I T13EX1
      T11LS1
      *6
      T13B1
      SPCFWD /FORCE MTF
      WCOV
      LOBR
      NOP /EXECUTE IOT; SHOULD NOT CLEAR
      SKTD /MTF;
      CMA /CHECK,
      COMPAR
      ER13B
/ I1= FAILING IOT
      JMP T13B+4
      JMS I T13EX2 /SUBTEST LOOP

/VERIFY THAT CLF DOES NOT CLEAR CA (OR ANY OTHER NON-STATUS
/REGISTER) SINCE "TUR" IS HIGH,
T13C, CLA CMA
      DCA GOOD
3434 7240
3435 3154

```

```

3436 7240      CLA CMA
3437 4475      LCAR
3440 4516      CLF
3441 4504      RCRAR
3442 4533      COMPAR
3443 7145      ER13C
          /GD= GOOD CA
3444 5236      JMP
          T13C+2
          /SUBTEST LOOP,

3445 4527      LOOP6
3446 5202      JMP
          T13A
          /*****LOOP 6*****

3447 7240      CLA CMA GOOD
3450 3154      DCA
3451 7240      CLA CMA
3452 4475      LCAR
3453 4543      WRITE
3454 4516      CLF
3455 4504      RCRAR
3456 4533      COMPAR
3457 7147      ER13D
          /GD= GOOD CA
          JMP
          T13D+2
          /SUBTEST LOOP,

3460 5251      /VERIFY SKCB, SKTD, SKEF, SKTR OR SOLE DO NOT GENERATE
          /BREAK REQUEST, CHECK VIA WC INCREMENT,
          T13E, JMS I T13EX1
          T11S1
          -5
          T13E1
          CLEAR2
          CLA CMA
          LWC
          SPDFWD
          T13E1, NOP
          3470 4545      RWC
          3471 0000      COMPAR
          3472 7000      ER13E
          3473 4503      /I1= FAILING IOT, GD= GOOD WC,
          3474 4533      JMP
          3475 7151      JMS I
          T13E+4
          T13EX2
          /SUBTEST LOOP

3500 4530      LOOP7
3501 5245      JMP
          T13D-2
          /*****LOOP 7*****

3502 5600      JMP I
3503 3122      T13EX1, TEST13
3504 3141      T13EX2, T11EX2
    
```





```

/TEST 14, FULL READ-COMPARE TEST,
/1, VERIFIES!
/1.1 NO READ-COMPARE ERROR WHEN MEM AND DB CONTAIN SAME
/1.2 READ-COMPARE ERROR IS GENERATED WHEN MEM AND DB
/1.3 CA DOES NOT INCREMENT WHEN R=C ERROR EXISTS,
/2, TESTS T14A AND T14B ARE RUN BY A LOCAL EXECUTIVE
WHICH GENERATES DATA FOR MEM AND DB, AND ALSO RUNS THE PORTION
OF THE TEST WHICH SETS UP CA, DB, RDCOMP FUNCTION, THE BREAK
REQUEST, THEN READS MS AND MASKS TO THE R/C BIT,
/3, ALL SUBTEST LOOPS CYCLE ON THE CURRENT DATA,
/4, LOOPS 6-8:
/4.1 LOOP 6 CYCLES ON T14A (NO R=C ERROR)
/4.2 LOOP 7 CYCLES ON T14B (R=C ERROR)
/4.3 LOOP 8 CYCLES ON T14C (CA NO INCREMENT)

```

3600	PAGE	
3600	TEST14, 0	
3601	CLEAR1	
3602	/VERIFY NO READ-COMPARE ERROR WHEN MEM AND DB CONTAIN SAME DATA,	
3603	/COUNT PATTERN USED,	
3604	T14A, DCA GOOD	
3605	JMS T14EX1	
3606	COMPAR	
3607	ER14A	
3610	/GD=GOOD MS MASKED; BD=REAL MS MASKED; OD=DATA IN MEM,	
3611	JMP T14EX2	
3612	JMS T14EX3	
3613	LOOP6	
3614	JMP T14A	/*****LOOP6*****
3615	/VERIFY READ-COMPARE ERROR IS GENERATED WHEN ONE BIT DIFFERS IN DB	
3616	/AND MEM, TAD K2	/VERIFY READ-COMPARE ERROR
3617	DCA GOOD	/WHEN ONE BIT DIFFERS IN DB
3618	JMS T14EX4	/AND MEM,
3619	COMPAR	/DATA IN DB UNDER "DB!",
3620	ER14B	
3621	/GD=GOOD MS MASKED; BD=REAL MS MASKED; OD=DATA IN MEM; DB=REAL DB	
3622	JMP T14EX5	/SUBTEST LOOP,
3623	JMS T14EX6	
3624	LOOP7	
3625	JMP T14B-2	/*****LOOP7*****
3626	/VERIFY CA DOES NOT INCREMENT WHEN READ-COMPARE ERROR EXISTS,	
3627	TAD XBUFFP	
3	1047	

```

3624 3154 DCA GOOD
3625 7240 CLA CMA
3626 3051 DCA XBUFF+1
3627 4524 CLEAR2
3630 4501 LDBR
3631 1047 TAD
3632 4475 LCAR
3633 1165 TAD PRGFLD
3634 4477 LCMR
3635 4542 RDCOMP
3636 4535 BREAK
3637 1047 TAD XBUFFP
3640 4475 LCAR
3641 7240 CLA CMA
3642 4501 LDBR
3643 4535 BREAK
3644 4504 RCAR
3645 4533 COMPAR
3646 7157 ER14C
/SET UP FOR NEXT CA INC.
/WITH R/C ERROR EXISTING,
T14C1, XBUFFP
/CD=600D CA, T14C1 /SUBTEST LOOP,
JMP

```

```

3650 1036 TAD K7400
3651 3051 DCA XBUFF+1
3652 3154 DCA GOOD
3653 1262 TAD +7
3654 3360 DCA T14EX8
3655 4524 CLEAR2
3656 1047 TAD XBUFFP
3657 4475 LCAR
3660 1005 TAD K3
3661 5366 JMP T14EX8+6
3662 3663 +1
3663 4533 COMPAR
3664 7161 ER14D
/VERIFY THAT CM10=11 INHIBIT READ=COMPARE CHECKS ON BITS
/0=3 OF DB AND MEM. 7400 IN MEM, 0000 IN DB, THERE SHOULD BE NO
/READ=COMPARE ERROR,
T14D, TAD
/CD=600D R/C ERROR BIT: BD=REAL R/C ERROR BIT
JMP T14D+5
SUBTEST LOOP,

```

```

3665 5255 JMP
3666 4531 LOOP8
3667 5221 JMP
3670 5600 JMP I TEST14
3671 0000 T14EX1, 0
3672 4347 JMS T14EX7
3673 1156 TAD OLD
3674 3051 DCA XBUFF+1
3675 1156 TAD OLD
3676 3160 DCA TXXTM2
3677 4360 JMS T14EX8
3700 5671 JMP I T14EX1
/ENTER FROM T14A,
/INITIALIZE DATA AND LOOP COUNTERS,
/PUT OLD IN MEM,
/PUT OLD IN LOC USED TO LOAD DB,
/SET UP CA, DB, RDCOMP, BREAK AND
/LOOK AT R/C BIT IN MS THEN GO TO

```

```

3701 0000 T14EX3, 0
3702 2156 ISZ
3703 5307 JMP
3704 2375 ISZ
3705 7410 SKP I
3706 5701 JMP I
3707 5273 JMP
3710 0000 T14EX4, 0
3711 4347 JMS
3712 1156 TAD
3713 3051 DCA
3714 1156 TAD
3715 1377 TAD
3716 3160 DCA
3717 1156 TAD
3720 0377 AND
3721 7104 CLL RAL
3722 7041 CIA
3723 1160 TAD
3724 3160 DCA

3725 4360 T14EX5, JMS
3726 5710 JMP I
3727 0000 T14EX6, 0
3730 2376 ISZ
3731 7410 SKP
3732 5337 JMP
3733 1377 TAD
3734 7104 CLL RAL
3735 3377 DCA
3736 5314 JMP
3737 7001 IAC
3740 3377 DCA
3741 2156 ISZ
3742 5346 JMP
3743 2375 ISZ
3744 7410 SKP
3745 5727 JMP I
3746 5355 JMP
3747 0000 T14EX7, 0
3750 1037 TAD
3751 3375 DCA
3752 3156 DCA
3753 7001 IAC
3754 3377 DCA
3755 1042 TAD
3756 3376 DCA
3757 5747 JMP I
3760 0000 T14EX8, 0
3761 4524 CLEAR2
3762 1160 TAD
3763 4501 LDBR
3764 1047 TAD
3765 4475 LCAR

/TEST 14A COMPAR SECTION,
/ENTER FROM END OF T14A,
/UPDATE OLD DATE,
/IF OLD NOT=0 RUN NEW OLD,
/IF OLD=0, HAS IT BEEN 0 TWICE?
/NO,
/YES, EXIT T14A,
/NO, RUN T14A AGAIN WITH NEW DATA,
/ENTER FROM T14B,
/INITIALIZE DATA AND LOOP COUNTERS,
/PUT OLD IN MEM.

/COMPLEMENT THE BIT IN OLD
/INDICATED BY CURRENT BIT IN
/T14BIT AND PUT MODIFIED OLD
/IN LOC USED TO LOAD DB,
/COMPLEMENTING THE BIT IS
/ACTUALLY AN EXCLUSIVE OR FUNCTION.)

/SET UP CA, DB, RCOMP, BREAK, AND LOOK
/AT R/C BIT IN MS; THEN GO TO T14B COMPAR,
/ENTER FROM END OF T14B,
/ALL 12 BITS BEEN COMPLEMENTED
/ONE AT A TIME?

/NO, MOVE BIT ONE LEFT,

/GO BACK AND COMPUTE NEW DB DATA,
/YES, PUT BIT IN POSITION 12,

/OLD 1 TO OLD, OLD = 0?

/YES, HAS IT BEEN 0 TWICE?

/YES, EXIT T14B,
/NO, RUN COMPLEMENTING BITS ON OLD,
/INITIALIZE LOOP COUNTERS AND
/DATE, SET UP FOR 2 RUNS
/FOR OLD (0192 DATA SETS),
/SET OLD TO 0 FOR A STARTER,
/SET BIT TO POSITION 12 FOR
/STARTER
/SET UP FOR 12 BIT ROTATES
/FOR USE IN T14B ONLY,
/EXIT
/T14A AND T14B TEST POSITION
/GENERATE "INITIALIZE,"
/GET DB DATA AND LOAD DB,
/PUT XBUFF IN CA,

```

3766 1165  
3767 4477  
3770 4542  
3771 4535  
3772 4505  
3773 0021  
3774 5760  
3775 0000  
3776 0000  
3777 0000

TAD  
LCMR  
RDCOMP  
BREAK  
RMSR  
AND  
JMP I  
K2  
T14EX8  
T14TM1: 0  
T14TM2: 0  
T14BIT: 0

PRGFLO  
/SET RDCOMP FUNCTION,  
/ONE BREAK REQUEST  
/HEAD MS AND MASK TO SEE  
/R/C ERROR BIT ONLY  
/GO TO COMPAR PORTION OF TEST



4055	6714	6714	/RMSR	
4056	5654	JMP I	SRMSR	
4057	4347	JMS	SKIPR	
4060	0000	SRCMR,		
4061	6715	0	/RCMR	
4062	5660	JMP I	SRCMR	
4063	4347	JMS	SKIPR	
4064	0000	SRFMR,		
4065	6716	0	/RFMR	
4066	5664	JMP I	SRFMR	
4067	4347	JMS	SKIPR	
4070	0000	SRDBR,		
4071	6717	0	/RDBR	
4072	5670	JMP I	SRDBR	
4073	4347	JMS	SKIPR	
4074	0000	S6720,		
4075	6720	0	/CONTROL JOT,	
4076	5674	JMP I	S6720	
4077	4347	JMS	SKIPR	
4100	0000	SSKEF,		
4101	6721	0	/SKEF	
4102	5700	JMP I	SSKEF	
4103	2300	ISZ	SSKEF	
4104	5700	JMP I	SSKEF	
4105	0000	SSKCB,		
4106	6722	0	/SKCB	
4107	5705	JMP I	SSKCB	
4110	2305	ISZ	SSKCB	
4111	5705	JMP I	SSKCB	
4112	0000	SSKTD,		
4113	6723	0	/SKTD	
4114	5712	JMP I	SSKTD	
4115	2312	ISZ	SSKTD	
4116	5712	JMP I	SSKTD	
4117	0000	SSKTR,		
4120	6724	0	/SKTR	
4121	5717	JMP I	SSKTR	
4122	2317	ISZ	SSKTR	
4123	5717	JMP I	SSKTR	
4124	0000	SCLF,		
4125	6725	0	/CLF	
4126	5724	JMP I	SCLF	
4127	4347	JMS	SKIPR	
4130	0000	SCLT,		
4131	6712	0	/CLT	
4132	5730	JMP I	SCLT	
4133	4347	JMS	SKIPR	
4134	0000	SSDLE,		
4135	6726	0	/SDLE	
4136	5734	JMP I	SSDLE	
4137	4347	JMS	SKIPR	
4140	0000	SSBRM,		
4141	6727	0	/SBRM	
4142	5740	JMP I	SSBRM	
4143	4347	JMS	SKIPR	

```

4144 0000 SODF, 0
4145 6201 SCDFI, /N
4146 5744 JMP I SCDF

SKIPER, 0
4147 0000 DCA ACLOC
4150 3170 TAD M4
4151 1040 TAD SKIPER
4152 1347 TAD SKIPER
4153 3347 DCA SKIPER
4154 7001 IAC
4155 1347 TAD SKIPER
4156 3155 DCA BAD
4157 1555 TAD I BAD
4160 3155 DCA BAD
4161 1747 TAD I SKIPER
4162 3347 DCA SKIPER
4163 1155 TAD BAD
4164 3375 DCA ,+11
4165 4534 ERROR
4166 6703 ER00C

/AN IOT SKIPPED WHICH SHOULD NEVER SKIP,
/BD=FAILING IOT CODE: AC=CONTENTS OF AC AT TIME OF FAILURE,
/EXECUTE IF SR5=1,
/IF NOT CHECK SR5,
/EXECUTE IF SR5=1,
/NO SCOPE LOOP, EXIT TO CORRECT
/POINT IN MAIN PROGRAM,
/SCOPE LOOP, SET UP AC, THIS
/MAY OR MAY NOT BE CORRECT CONDITION,
/IOT CODE, NO FAILURE, CHECK SR5,
/FAILURE, CALL ERRORS,

4167 5374 JMP ,+5
4170 4526 L00P5 ,+3
4171 5374 JMP ACLOC
4172 1170 TAD SKIPER
4173 5747 JMP I ACLOC
4174 1170 TAD 0
4175 0000 JMP ,+6
4176 5370 JMP ,+12
4177 5365

```



/EXECUTIVE

/USING A TEST STATUS WORD, "TSTAT", EXEC DETERMINES WHICH TESTS ARE TO BE RUN. EACH BIT IN "TSTAT" REPRESENTS ONE TEST. (BIT N REPRESENTS TEST N+1 IN OCTAL.)  
 /"TSTAT" IS AUTOMATICALLY SET TO 7777 WHEN THE PROGRAM IS /STARTED AT 200, WITH SR0=0. IF SR0=1 AT TEST START TIME, /USER MUST SELECT TESTS WITH MONITOR. MONITOR THEN SETS THE /SELECTED TEST BITS AND ENTERS EXEC AT REX. ALL TESTS /WILL BE RUN THE NUMBER OF TIMES SPECIFIED BY THEIR /RESPECTIVE TALLY CONSTANTS UNLESS QUICK VERIFY IS SET. /WHEN ALL TESTS ARE COMPLETE, EXEC GOES TO MONITOR.

EXEC	RIF	PAGE	INSTRUCTION
4200	DCA	4200	/GET INSTRUCTION FIELD;
4201	TAD	4201	/MODIFY HOMING CDF ROUTINE;
4202	TAD	4202	
4203	DCA I	4203	/SET UP INTERRUPT LINKAGE;
4204	CDF	4204	/PUT RMF IN 1;
4205	TAD	4205	
4206	DCA I	4206	/JMP 1 3 IN 2;
4207	TAD	4207	/INTSEV IN 3;
4210	DCA I	4210	
4211	DCA I	4211	/SET BACK TO CURRENT FIELD;
4212	TAD	4212	/CLEAR ERROR EXIT FLAG;
4213	DCA I	4213	/SET THE TTD SOFT FLAG;
4214	JMS I	4214	
4215	DCA CMA	4215	
4216	DCA	4216	
4217	S&L	4217	
4220	JMP	4220	
4221	JMS I	4221	
4222	MSG1	4222	/PRINT TITLE MESSAGE
4223	CLA CMA	4223	/SET TEST IN PROGRESS FLAG
4224	DCA	4224	/FIND OUT HOW MANY MEM FIELDS
4225	JMS I	4225	/GET SR0 AND
4226	DCA	4226	/SAVE;
4227	LAS	4227	/TELL USER THAT SR0 IS NOW A
4230	DCA	4230	/QUICK VERIFY SWITCH
4231	JMS I	4231	/GET OLD SR0
4232	MSG9	4232	/GO DIRECTLY TO MONITOR?
4233	TAD	4233	/YES.
4234	SPA CLA	4234	/NO. SET UP TO RUN ALL TESTS,
4235	JMP	4235	/PUT AC IN TEST STATUS;
4236	CLA CMA	4236	/CLEAR ERROR EXIT FLAG
4237	DCA	4237	/SET TEST IN PROGRESS
4240	DCA	4240	
4241	DCA	4241	
4242	CLA CMA	4242	
4243	DCA	4243	
4244	DCA	4244	
4245	TAD	4245	
4246	DCA	4246	

REX,  
 EXECCL2, TAD

4247	1365	TAD	TSTP	/GET TEST STARTING ADDS, TABLE POINTER
4250	3355	DCA	TSTPP	/GET TEST TALLY TABLE POINTER
4251	1367	TAD	TALP	/SET TEST NUMBER TO '0
4252	3356	DCA	TALPP	/GET TEST STATUS AND PUT IN
4253	3177	DCA	TSTNUM	/TEMP,
4254	1176	TAD	TSTAT	/CHECK FOR A TEST BIT SET
4255	3357	DCA	TSTAEX	/SAVE TEST STATUS ROTATED;
4256	1357	TAD	TSTAEX	/UPDATE POINTERS
4257	7004	RAL	TSTAEX	/UPDATE TEST NUMBER
4260	3357	DCA	TSTPP	/RUN THIS TEST?
4261	2355	ISZ	TSTPP	/NO
4262	2356	ISZ	TALPP	/YES TEST PROGRESS REPORT?
4263	2177	ISZ	TSTNUM	
4264	7420	SNL	TSTNUM	
4265	5315	JMP	NOTSRN	
4266	7604	LAS	K2000	
4267	0033	AND		
4270	7640	SZA	CLA	
4271	5277	JMP	NOTSRP	/NO
4272	1177	TAD	TSTNUM	/YES, GET TEST NUMBER
4273	4463	JMS	I	/INSERT IN MSG AND PRINT
4274	3760	DCA	I	
4275	4461	JMS	I	
4276	7213	MSG2	AMG8EP	
4277	1756	NOTSPR,	TAD	/GET THIS TESTS TALLY AND SAVE
4300	3361	DCA	TALPP	/GET TEST STARTING ADDRESS
4301	1755	TAD	TSTAL	
4302	3362	DCA	TSTPP	
4303	4762	JMS	I	/RUN TEST
4304	7604	LAS	I	/LOOP ON CURRENT TEST
4305	0021	AND	K2	
4306	7640	SZA	CLA	
4307	5303	JMP	EXOMT	/YES RUN TEST AGAIN
4310	7604	LAS	EXOMT	/SR0 SET? (QUICK VERIFY)
4311	7710	SPA	CLA	/NO, GO CHECK TEST TALLY
4312	5315	JMP	NOTSRN	/YES SEE IF ALL TESTS RUN
4313	2361	ISZ	TSTAL	/NO DECREASE TALLY, TEST DONE?
4314	5303	JMP	EXOMT	/NO RUN AGAIN
4315	2354	ISZ	EXTAL	/YES 12 TESTS CHECKED?
4316	5256	JMP	EXECL1	/NO CHECK FOR NEXT TEST
4317	3177	DCA	TSTNUM	
4320	7604	LAS		/YES, LOOP ON ALL SELECTED TESTS
4321	0020	AND	K1	
4322	7650	SNA	CLA	
4323	5336	JMP	NOEXLP	/NO
4324	2346	ISZ	PASCNT	/YES UPDATE PASS COUNTER
4325	1346	TAD	PASCNT	/INSERT IN MSG AND
4326	4463	JMS	I	/PRINT
4327	3763	DCA	I	
4330	1464	TAD	M4P2P	
4331	3764	DCA	I	
4332	4461	JMS	I	
4333	7233	MSG4	AMG8EP	
4334	4465	JMS	I	/BEL

4335	5245	JMP	EXECCL2	
4336	4461	JMS I	AMGBEP	/RUN ALL OVER AGAIN
4337	7220	MS03		/PRINT DONE MESSAGE,
4340	3175	DCA	ACTFLG	/CLEAR TEST IN PROGRFSS,
4341	7240	CLA CMA		
4342	3173	DCA	TYOFLG	
4343	5462	JMP I	MONITP	/GO TO MONITOR
4344	5403	KJMPIS, 5403		
4345	5400	INTSEP, INTSEV		
4346	0000	PASCNT, 0		
4347	6244	KRMF, RMF		
4350	4145	SCDFIP, SCDFI		
4351	0001	P1, 1		
4352	0002	P2, 2		
4353	0003	P3, 3		
4354	0000	EXTAL, 0		
4355	0000	TSTPP, 0		
4356	0000	TALPP, 0		
4357	0000	TSTAEX, 0		
4360	7216	M2P1P, M2P1		
4361	0000	TSTTAL, 0		
4362	0000	EXTEMP, 0		
4363	7237	M4P2P, M4P2		
4364	7236	M4P1P, M4P1		
4365	6562	TSTP, TST=1		
4366	0203	TET1P, TEST01		
4367	6576	TALP, TAL=1		

/MONITOR

/1. ENTERED UNDER ONE OF THE FOLLOWING CONDITIONS:  
 /1.1 PROGRAM STARTED AT 202 WITH SR0=1,  
 /1.2 ERROR OCCURS WITH SR2=0,  
 /1.3 ANY FATAL ERROR OCCURS,  
 /1.4 "ALTMODE" IS STRUCK ON THE KEYBOARD AFTER TEST START,  
 /1.5 SELECTED TESTS ARE RUN TO COMPLETION.

4400	PAGE		
4401	MONIT,	JMS I	/INTERRUPT SYSTEM OFF;
4402		DCA	/SAVE SUBROUTINE ONFO;
4403		JMS I	
4404		MSG8	/PRINT "C";
4405		DCA	/CLEAR TEST STATUS TEMP;
4406		JMS I	/GET KEYBOARD;
4407		-303	
4410		CS	/C STRING;
4411		-324	
4412		TS	/T STRING;
4413		-305	
4414		ES	/E STRING;
4415		-304	
4416		D5	/"D" LOOK AT MEMORY;
4417		0	
4420	CS,	DCA	/"C" OR "EX"; PUT AC IN TEMP;
4421		TAD	/IF NO TEST IN PROGRESS; SYNTAX
4422		SMA CLA	/ERROR;
4423		JMP I	/GET KEYBOARD;
4424		JMS I	/"E";
4425		-215	
4426		+2	
4427		0	
4430		JMS I	/"C" OR "EX";
4431		JMS I	/RESTORE SUBROUTINES;
4432		TAD	/GET TEMP AND PUT IN EXIT
4433		DCA	/GLAG;
4434		JMP I	/EXIT TO INTERRUPT SERVICE;
4435	TS,	JMS I	/T STRING; GET KEYBOARD;
4436		-322	
4437		TR	/"TR"
4440		-301	
4441		TA	/"TA"
4442		1	
4443		+2	
4444		0	
4445		CLL RAL	/GET FIRST NUMBER; SCALE
4446		RTL	/AND SAVE;
4447		DCA	
4450		JMS I	/GET NEXT NUMBER
4451		1	
4452		+2	
4453		2	

4454	1335	TAD	TSTEM1	/ADD IWO NUMBERS AND CONVERT
4455	7041	CIA		/SUM TO ISZ SENSITIVE COUNTER,
4456	3335	DCA	TSTEM1	/DETERMINE TEST BIT,
4457	7120	CLL CML		
4460	7010	RAH		
4461	2335	ISZ	TSTEM1	/SAVE TEST BIT,
4462	5260	JMP	.#2	/EXCLUSIVE OR NEW TEST BIT
4463	3335	DCA	TSTEM1	/WITH OLD TEST STATUS,
4464	1335	TAD	TSTEM1	
4465	0337	AND	TSTATM	
4466	7104	CLL RAL		
4467	7041	CIA		
4470	1335	TAD	TSTEM1	/SAVE NEW TEST STATUS,
4471	1337	TAD	TSTATM	
4472	3337	DCA	TSTATM	/GET INPUT,
4473	4743	JMS I	LISNP	/"TNN#"
4474	7563	-215		/"TNNI#"
4475	4513	TCR		/GO BACK TO T STRING,
4476	7454	-324		
4477	4435	TS		
4500	0000	0		
4501	7240	CLA CMA	TRACE	/"TR" TRACE REQUESTED,
4502	3171	DCA	LISNP	/SET TRACE FLAG,
4503	4743	JMS I		/GET KEYBOARD,
4504	7475	-303		
4505	4420	CS		/"TRC"
4506	7454	-324		
4507	4435	TS		/"TRT"
4510	7473	-305		/"TRE"
4511	4527	ES		
4512	0000	0		/NULL MAGTAPE INTERRUPTS, ION,
4513	4523	CLEAR1		
4514	4466	JMS I	CRLF	/PUT TEST STATUS IN AC,
4515	1337	TAD	TSTATM	/GO TO "EXEC."
4516	5736	JMP I	REXP	/SET ALL TESTS,
4517	7240	CLA CMA	TSTATM	
4520	3337	DCA	LISNP	/GET KEYBOARD,
4521	4743	JMS I		
4522	7454	-324		/"TAT"
4523	4435	TS		/"TA#"
4524	7563	-215		
4525	4513	TCR		/E STRING,
4526	0000	0		/"EX"
4527	4743	JMS I	LISNP	
4530	7450	-330		
4531	4533	.#2		
4532	0000	0		/SET ERROR EXIT FLAG,
4533	7240	CLA CMA	CS	/EXIT VIA C STRING,
4534	5220	JMP		
4535	0000	0		
4536	4240	TSTEM1,		
4537	0000	REXP,		
4540	6000	TSTATM,		
4541	6021	SAVEPP,		
		RESTPP,		
		RESTP,		

4542 4673 GUESP, QUES  
 4543 4677 LISNP, LISN  
 4544 5444 INTOKP, INIOK

4602	PAGE		
4600	TAD	M4	/SET FOR 4 NUMBERS,
4601	DCA	DS2	
4602	JMS	LISN	/LOOK FOR FIELD NUMBER;
4603	1		
4604	,+2		
4605	0		
4606	CLL	RAL	/SCALE AND SAVE,
4607	RTL		
4610	DCA	DS1	
4611	JMS	LISN	/LOOK FOR ;
4612	-272		
4613	,+2		
4614	0		
4615	TAD	DS1	/COMPUTE GDF INSTRUCTION,
4616	TAD	KCDF	
4617	DCA	DLF	
4620	DCA	DS1	/CLEAR TEMP FOR NEXT 4 INPUTS,
4621	JMS	LISN	/GET KEYBOARD,
4622	1		
4623	,+2		/OCTAL NUMBER,
4624	0		
4625	TAD	DS1	/ADD TEMP,
4626	ISZ	DS2	/4TH NUMBER?
4627	SKP		
4630	JMP	,+5	
4631	CLL	RAL	/NO, MOVE 3 PLACES LEFT,
4632	RTL		/STORE,
4633	DCA	DS1	
4634	JMP	DLF2	/GO TO NEXT NUMBER,
4635	DCA	DS1	/YES, 4TH NUMBER, SAVE ADDRESS,
4636	DCA	/N	/CHANGE TO REQUESTED DATA FIELD,
4637	DCF	DS1	/GET CONTENTS,
4640	TAD	I	/CHANGE BACK TO THIS PROGRAM FIELD,
4641	JMS	I	/EDIT AND INSERT IN
4642	JMS	I	/MESSAGE,
4643	DCA	I	
4644	TAD	I	
4645	DCA	I	
4646	JMS	I	
4647	MSG11		
4650	JMS	LISN	/PRINT MESSAGE,
4651	-215		
4652	DCR		
4653	-212		
4654	,+2		
4655	0		
4656	TAD	K215	/DLF", DUMP NEXT LOCATION,
4657	JMS	I	
4660	ISZ	TYPEP	
4661	JMP	DS1	
4662	DCA	DLF	
4663	TAD	DLF	

4662	1023	TAD	K10	
4663	3236	DCA	DLF	
4664	5236	JMP	DLF	
4665	4466	JMS I	CRLEP	/D*. LOOK FOR NEW COMMAND.
4666	5676	JMP I	MONP	
4667	0000	0		
4670	2000	0		
4671	7310	M11P1,	M11P1	/TYPE "P*", RING BELL.
4672	7311	M11P2,	M11P2	/RESTART MONITOR.
4673	4461	JMS I	AMG8P	/GET KEYBOARD AND CHECK SYNTAX.
4674	7241	MSG6		
4675	5676	JMP I	MONP	
4676	4403	MONIT+3		
4677	0000	0		
4700	6031	KSF		
4701	5300	JMP	"=1	
4702	6036	KR8		
4703	3346	DCA	LISN11	/SAVE INPUT
4704	1346	TAD	LISN11	/ECHO.
4705	4470	JMS I	TYPEP	/DO NOT CHECK "SPACE" IN SYNTAX.
4706	1346	TAD	LISN11	
4707	1347	TAD	M240	
4710	7650	SNA	CLA	
4711	5300	JMP	LISN+1	
4712	1677	TAD I	LISN	/GET COMPATOR.
4713	7450	SNA		
4714	5273	JMP	QUES	/IF 0, SYNTAX ERROR.
4715	7500	SMA		
4716	5327	JMP	LISNUM	/IF >0, LOOK FOR OCTAL NUMBER.
4717	1346	TAD	LISN11	/MATCH?
4720	7640	SZA	CLA	
4721	5324	JMP	LISN2	/NO.
4722	3346	DCA	LISN11	/YES, FAKE TEMP CLEAR AND
4723	5340	JMP	LISN3	/EXIT.
4724	2277	ISZ	LISN	/UPDATE CALL POINTERS.
4725	2277	ISZ	LISN	
4726	5312	JMP	LISN1	/GET NEXT COMPATOR.
4727	7200	CLA		/OCTAL NUMBER REQUIRED.
4730	1346	TAD	LISN11	
4731	1044	TAD	M260	
4732	7710	SPA	CLA	
4733	5324	JMP	LISN2	/NOT AN OCTAL NUMBER
4734	1346	TAD	LISN11	/YES, UPDATE CALL.
4735	1045	TAD	M270	/GET RETURN ADDRESS.
4736	7700	SMA	CLA	
4737	5324	JMP	LISN2	
4740	2277	ISZ	LISN	
4741	1677	TAD I	LISN	
4742	3277	DCA	LISN11	/GET 0 OR OCTAL NUMBER CODE
4743	1346	TAD	K7	/MASK TO LOW ORDER 3 BITS.
4744	2022	AND		
4745	5677	JMP I	LISN	/EXIT.
4746	2000	0		
4747	7540	M240,		"=240





```

5020 5227 JMP ,+7 ERRPTR
5021 1745 TAD I
5022 7004 RAL
5023 7700 SMA CLA
5024 5227 JMP
5025 2200 ISZ
5026 3174 DCA
5027 2200 ISZ
5030 5600 JMP I

5031 4461 FATEHR, JMS I AMGSEP
5032 7243 MSG7
5033 4465 JMS I BELL
5034 1372 TAD K605
5035 4253 JMS PRNT
5036 3175 DCA ACTFLG
5037 5462 JMP I MONITP
5040 0000 OPRNT, 0
5041 7604 LAS
5042 0032 AND
5043 7640 SZA CLA
5044 5640 JMP I
5045 1371 TAD K400
5046 4253 JMS PRNT
5047 7240 CLA CMA
5050 1345 TAD
5051 3345 DCA
5052 5640 JMP I

5053 0000 PRNT, 0
5054 3746 DCA I ERMP1P
5055 1745 TAD I ERRPTR
5056 4522 BSW
5057 0365 AND
5060 4463 JMS I EDITP
5061 3747 DCA I ERMP2P
5062 1745 TAD I ERRPTR
5063 4522 BSW
5064 0373 AND
5065 1025 TAD K7700
5066 3750 TAD K400
5067 1200 DCA I ERMP3P
5070 4463 TAD I ERRORS
5071 3753 JMS I EDITP
5072 1464 DCA I ERMP5P
5073 3752 TAD I EDTEMP
5074 2345 DCA I ERMP4P
5075 1041 ISZ ERRPTR
5076 3354 DCA M5
5077 1042 TAD M14
5100 3355 DCA M14
5101 1362 TAD PRCNT1
5102 3357 DCA PRCNT2
5103 1356 TAD SYMPTP
5104 3360 DCA SYMPTP

```

```

/NO,
/YES, DOES THIS ERROR REQUIRE
/"EX." SERVICE?

/NO, LEAVE EXITFL SET,
/YES, MOVE POINTER TO RECYCLE JUMP,
/CLEAR "EX." FLAG,
/MOVE POINTER ONE MORE POSITION,
/EXIT,

/FATAL ERROR, PRINT MSG.

/PUT "FE" IN AC,
/PRINT ERROR MESSAGE,
/CLEAR TEST IN PROGRESS FLAG,
/GO TO MONITOR,
/PRINTOUT?

/NO, EXIT,
/YES, PUT "ER" IN AC AND
/GENERATE ERROR PRINTOUT,
/BACK BIAS ERROR TABLE POINTER,
/TO POINT TO STAT1,
/EXIT.

/GENERATE ERROR MESSAGE,
/SAVE "FE" OR "ER",
/GET TEST NUMBER FROM
/STAT1 AND PUT IN ERROR
/MESSAGE,

/GET SUBTEST LETTER FROM STAT1
/AND PUT IN ERROR MESSAGE
/AFTER APPENDING "SPACE" CODE,

/EDIT RETURN JUMP FOR "PC" PRINTOUT,

/MOVE ERROR TABLE POINTER TO STAT2,
/SET COUNTER FOR 5 SYMBOLS
/BEFORE CR=LF,
/SET COUNTER FOR 12 SYMBOLS
/MAXIMUM,
/GET SYMBOL ROUTINE TABLE
/POINTER AND PUT IN TEMP,
/GET ERROR MESSAGE INSERT
/POINTER,

```

5105	STL RAR		
5106	DCA	PRNTK	/INITIALIZE TO SPACE
5107	LAS		/COMPLETE DUMP?
5110	AND	K230	
5111	SNA CLA		
5112	JMP	,+3	/YES,FAKE STAT2 FULL.
5113	CLA CMA		
5114	SKP		
5115	TAD I	ERRPTR	/GET STAT2 AND PUT IN TEMP.
5116	DCA	WHAT	/BIT BY BIT INSPECT STAT2 FOR
5117	TAD	WHAT	/SYMBOL PRINTOUTS, AS A
5120	RAL		/BIT IS SENSED SFT, GO TO
5121	DCA	WHAT	/SYMBOL ROUTINE AND INSERT
5122	SNL		/SYMBOL AND DATA IN
5123	JMP	PRNTCK	
5124	TAD I	SYMBOL	
5125	DCA	SYMADR	
5126	JMS I	SYMADR	
5127	ISZ	PRCNT1	
5130	JMP	,+3	/5 SYMBOLS PRINTED?
5131	TAD	K4300	/YES, INSERT 1 CR=LF IN SPACING.
5132	SKP		//NO, INSERT 1 SPACE.
5133	STL RAR		
5134	DCA	PRNTK	/UPDATE SYMBOL TABLE POINTER.
5135	ISZ	SYMBOL	/11 SYMBOLS CHECKED?
5136	ISZ	PRCNT2	/NO, CHECK AGAIN.
5137	JMP	PRNTLP	/YES, PUT CR=LF AND END
5140	TAD	K4300	/MSG CODE IN ERROR MSG;
5141	DCA I	PRMPT	/PRINT ERROR MESSAGE.
5142	JMS I	AMG8EP	
5143	ERMSG		/EXIT
5144	JMP I	PRNT	/TEMP FOR STATUS WORD POINTER;
5145	0		/ERROR MESSAGE POINTERS; ERROR TYPE,
5146	ERMP1		/I TEST NUMBER
5147	ERMP2		/I SUBTEST LETTER=SPACE.
5147	ERMP3		
5150	ERMP3P		
5151	SYMADR, 0		/HIGH ORDER PC.
5152	ERMP4P		/I LOW ORDER PC.
5153	ERMP5P		/SYMBOL POINTER.
			/5 SYMBOL COUNTER.
			/12 SYMBOL COUNTER.
			/SYMBOL TABLE POINTER, START POINTER,
			/ERROR MESSAGE INSERTER TEMP,
			/SYMBOL TABLE POINTER TEMP,
			/ERROR MESSAGE INSERT POINTER,
			/SPACING CONSTANT,
5154	PRCNT1, 0		
5155	PRCNT2, 0		
5156	ERMP5P, 0		
5157	ERMP5P+1		
5160	SYMBOL, 0		
5160	PRMPT, 0		
5161	PRNIK, 0		
5162	SYMPTP, 6663		
5163	WHAT, 0		/TEMP FOR STAT2 ROTATES.
5164	CHALTP, 0		/POINTER TO RETURN TO MONITOR CHECK.
5165	K17, 17		
5166	K4300, 4300		
5167	K4343, 4343		
5170	K4352, 4352		

5171	0522	K522,	522						
5172	0605	K605,	605						
5173	7700	K7700,	7700						
5200	0000	QHALT,	0	PAGE					/RETURN TO MONITOR?
5201	7604	LAS							
5202	0366	AND		K1000					
5203	7640	SZA	CLA						
5204	5600	JMP	I	QHALT					
5205	6002	IOF							
5206	1213	TAD		HALTGP					
5207	3000	DCA		0					
5210	5462	JMP	I	MONITP					
5211	7200	HALTC,	CLA						
5212	5600	JMP	I	QHALT					
5213	5211	HALTCP,	HALTC						
5214	0000	/SYMBOL	ROUTINES,						
5215	1154	GO,	0	GOOD,					
5216	4330	TAD		SYMSEV					
5217	0407	JMS							
5220	5614	0407		JMP	I	GO			
5221	0000	0							
5222	1155	TAD		BAD					
5223	4330	JMS		SYMSEV					
5224	1402	0402							
5225	5621	JMP	I	BD					
5226	0000	0							
5227	1156	TAD		OLD					
5230	4330	JMS		SYMSEV					
5231	0417	0417							
5232	5626	JMP	I	OD					
5233	0000	0							
5234	4503	RNCR							
5235	4330	JMS		SYMSEV					
5236	0327	0327							
5237	5633	JMP	I	WC					
5240	0000	0							
5241	4504	RCAR							
5242	4330	JMS		SYMSEV					
5243	0103	0103							
5244	5640	JMP	I	CA					
5245	0000	0							
5246	4506	RCMR							
5247	4330	JMS		SYMSEV					
5250	1503	1503							
5251	5645	JMP	I	CM					
5252	0000	0							
5253	4507	RFSR							
5254	4330	JMS		SYMSEV					
5255	2306	2306							
5256	5652	JMP	I	FS					
5257	0000	0							

AND, TURN INTERRUPT OFF,  
 /PUT CONTINUE EXIT IN 0  
 /SO "C" OR "EX" EXITS TO  
 /HALTC, GO TO MONITOR,  
 /CONTINUE,  
 /EXIT,  
 /HALTC POINTER;

/GOOD,  
 /GET GOOD,  
 /"GD",  
 /GET BAD,  
 /"BD",  
 /GET OLD,  
 /"OD",  
 /GET WC,  
 /"WC",  
 /GET CA,  
 /"CA",  
 /GET CM,  
 /"CM",  
 /GET FS,  
 /"FS"

5260	4575	RMSR		/GET MS
5261	4330	JMS	SYMSEV	
5262	2315	2315		/"MS"
5263	5657	JMP I	MS	
5264	0000	0		
5265	4510	RDBR		/GET DB
5266	4330	JMS	SYMSEV	
5267	0204	0204		/"DB"
5270	5664	JMP I	DB	
5271	0000	0		
5272	1170	TAD	ACLOC	
5273	4330	JMS	SYMSEV	
5274	0301	0301		
5275	5671	JMP I	AC	
5276	0000	0		
5277	1703	TAD I	IOT1P	
5300	4312	JMS	I112	
5301	6111	6111		
5302	5676	JMP I	I1	
5303	0566	IOT1P,		
5304	0000	I2,		
5305	1711	TAD I	IOT2P	
5306	4312	JMS	I112	
5307	6211	6211		
5310	5704	JMP I	I2	
5311	0571	IOT2P,		
5312	0000	I112,		
5313	0370	0		
5314	1366	AND	K377	
5315	1316	TAD	K1000	
5316	0000	DCA	.+1	
5317	7001	0		
5320	3316	IAC	.+2	
5321	1712	DCA	I112	
5322	3325	TAD I	.+3	
5323	1716	DCA	.+5	
5324	4330	TAD I	SYMSEV	
5325	0000	JMS		
5326	2312	0		
5327	5712	ISZ	I112	
5330	0000	JMP I	I112	
5331	3365	SYMSEV,		
5332	1763	DCA	SYMTEM	
5333	3364	TAD I	ERMPP	
5334	1730	DCA	ERMSYM	
5335	2377	TAD I	SYMSEV	
5336	1776	AND	(77	
5337	3764	TAD I	(PRNTK	
5340	2364	DCA I	ERMSYM	
5341	1730	ISZ	ERMSYM	
5342	0375	TAD I	SYMSEV	
5343	1357	AND	(7700	
5344	3764	TAD	K72	
		DCA I	ERMSYM	

/ENTER FROM I1 OR I2,  
 /CONVERT INST TO TAD POINTER,  
 /INTO TAD POINTER,  
 /STORE TAD POINTER HERE,  
 /POINTER +1 WHICH IS ADDRESS  
 /OF IOT CODE,  
 /GET SYMBOL CODE,  
 /PUT IN CALL TO SYMSEV;  
 /GET IOT CODE,  
 /GO TO SYMBOL SERVICE,  
 /SYMBOL CODE GOES HERE,  
 /UPDATE RETURN TO I1 OR I2,  
 /RETURN,  
 /COMMON SYMBOL ROUTINE SERVICE,  
 /SAVE DATA,  
 /GET ERROR MESSAGE CURRENT POINTER,  
 /PUT ON THIS PAGE OF MEMORY,

5345	2364	ISZ	ERMSYM	/DATA,
5346	1365	TAD	SYNTEM	/GET DATA,
5347	4463	JMS I	EDITP	/EDIT,
5350	3365	DCA	SYNTEM	/SAVE LOW ORDER,
5351	1464	TAD I	ERTEMP	/GET HIGH ORDER AND INSERT,
5352	3764	DCA I	ERMSYM	/UPDATE POINTER TO LOW ORDER
5353	2364	ISZ	ERMSYM	/DATA, GET LOW ORDER AND INSERT,
5354	1365	TAD	SYNTEM	
5355	3764	DCA I	ERMSYM	/UPDAT POINTER TO SPACING,
5356	2364	ISZ	ERMSYM	/PUT CURRENT POINTER ON
5357	1364	TAD	ERMSYM	/"ERRORS" MEMORY PAGE,
5360	3763	DCA I	ERMPP	/UPDATE RETURN,
5361	2330	ISZ	SYNSEV	/EXIT,
5362	5730	JMP I	SYNSEV	
5363	5160	PRMPT		
5364	0000	ERMSYM, 0		
5365	0000	SYNTEM, 0		
5366	1000	K1000, 1000		
5367	0072	K72, 72		
5370	0377	K377, 377		

/GENERAL INTERRUPT HANDLER,

/CHECKS FOR TIO AND TTI INTERRUPTS AND SERVICES SAME,  
 /CHECKS APPLICABILITY OF EF AND MTF INTERRUPTS AND  
 /GOES TO SPECIFIED EF OR MTF INTERRUPT HANDLER IF  
 /SPECIFIED IN INTEF AND INTMTF,

ADDRESS	INSTRUCTION	OPERAND	COMMENT
5375			
7700			
5376			
5161			
5377			
2077			
5400			
5420	INTSEV,	OCA	SAVEAC
5401		RAR	/SAVE AC AND LINK,
7010		DCA	
3163		DCA	SAVEL
5403		CDF	00
6201		TAD I	P0
5404		DCA	0
1662		JMS I	SCDFP
5405		JMS I	SCDFP
3000		TSP	
5406	INTAGN,	JMP	,+5
4457		TCF	
5407		CLA	CMA
6041		DCA	
5410		JMP	
5215		KSF	
5411		JMP	
6042		KRB	
5412		TAD	M233
7240		SNA	
3173		JMP I	MONITP
5413		TAD	M142
5414		SNA	
5244		JMP I	MONITP
5415		TAD	M1
6031		SNA	CLA
5416		JMP I	MONITP
5232		JMP	
5417		TAD	
6036		SNA	CLA
5420		JMP	
1257		TAD	
5421		SNA	
7450		JMP I	MONITP
5422		TAD	M142
5462		SNA	
5423		JMP I	MONITP
1260		TAD	
5424		SNA	CLA
7450		JMP I	MONITP
5425		TAD	
5462		SNA	CLA
5426		JMP I	MONITP
1261		TAD	
7650		SNA	CLA
5427		JMP I	MONITP
5430		JMP	
5462		TAD	
5431		SNA	CLA
5244		JMP I	MONITP
5432	INTNGA,	TAD	
1164		SNA	CLA
5433		JMP I	MONITP
7640		TAD	
5434		SNA	CLA
5564		JMP I	MONITP
5435		TAD	
1166		SNA	CLA
5436		JMP I	MONITP
7640		TAD	
5437		SNA	CLA
5566		JMP I	MONITP
5440		TAD	
1000		DCA	BAD
5441		ERROR	
3155		FE00A	
5442			
4534			
5443			
6677			
5444			
7300			

/ILLEGAL INTERRUPT. THE INTERRUPT CAUSE MAY BE THE TMP-E OR ANY  
 /DEVICE EXCEPT THE TTY. THIS ERROR IS FATAL AND KILLS ALL TEST  
 /SELECTIONS.  
 /BD=CONTENTS OF LOCATION 0; ALL OTHERS ARE CURRENT VALUES AND  
 /MAY NOT APPLY.  
 INTOK, CLA CLL

```

5445 1163 TAD SAVEI /AND EXIT;
5446 7004 RAL
5447 1172 TAD INTELG
5450 7640 SZA CLA
5451 5254 JMP +3
5452 1162 TAD SAVEAC
5453 5400 JMP I 0
5454 1162 TAD SAVEAC
5455 6001 ION
5456 5400 JMP I 0
5457 7545 M233, -233
5460 7636 M142, -142
5461 7777 M1, +1
5462 0000 P0, 0

```

```

/EF AND MTF INTERRUPT SERVICE ROUTINES SPECIFIER,
/ENTER WITH
/JMS I INSRSP
/ADDRESS OF EF SERVICE PUT IN INTEF PUT IN INTMTF
/ADDRESS OF MTF SERVICE PUT IN INTMTF

```

```

INSRS, 0
CLA CLL CMA /SET ACTIVE FLAG;
DCA ACTFLG
TAD I INSRSP
DCA INTEF
ISZ INSRSP
TAD I INSRSP
DCA INTMTF
TAD INTEF
TAD INTMTF
SZA CLA INSRSP
JMP GOOD
DCA GOOD
LCMR
RCMR
COMPAR
ER00B
/CM SHOULD CONTAIN 0000 BUT DOES NOT,
/GD=GOOD CMT,
JMP +4
INSRSF, ISZ INSRSP
JMP I INSRSP

```

```

5463 0000
5464 7340
5465 3175
5466 1663
5467 3164
5470 2263
5471 1663
5472 3166
5473 1164
5474 1166
5475 7640
5476 5305
5477 3154
5500 4477
5501 4506
5502 4533
5503 6701

5504 5300
5505 2263
5506 5663

```

/EXECUTIVE ADDENDUM,  
EXECFX, 0 /ENQUIRE ABOUT EXISTING  
MSG10 /EXTENDED MEMORY FIELDS  
KSF /FOR THE CA MEMORY REFERENCE  
M260 /TEST,

5507	0000		
5510	4461	JMS I	AMGSEP
5511	7273	MSG10	
5512	6031	KSF	
5513	5312	JMP	M260
5514	6036	KRB	
5515	3157	DCA	TXXTM1
5516	1157	TAD	TXXTM1
5517	4470	JMS I	TYPEP
5520	1157	TAD	TXXTM1
5521	1044	TAD	M260
5522	7710	SPA	CLA
5523	5310	JMP	EXECFX+1
5524	1157	TAD	TXXTM1
5525	1045	TAD	M270
5526	7700	SMA	CLA
5527	5310	JMP	EXECFX+1
5530	1157	TAD	TXXTM1
5531	0022	AND	K7
5532	7001	IAC	
5533	7041	CIA	
5534	3167	DCA	MEMFLD
5535	4466	JMS I	ORLFP
5536	5707	JMP I	EXECFX



```

/COMPARE AND ERROR DETECTION ROUTINE.
/ENTER WITH "BAD" IN AC.
/USE THE FOLLOWING CALLING SEQUENCE:
/COMPAR      CALL COMPARE,
/ERNNX      ADDRESS OF ERROR, (ERROR CODE);
/JMP        LOOP JUMP
/.....CONTINUATION      TEST CONTINUE;

/COMPARE DOES THE FOLLOWING:
/1. COMPARE "GOOD" AND "BAD".
/2. IF NOT EQUAL APPEARS TO GO TO ERRORS FROM POINT IN TEST
/   WHERE ERROR OCCURRED.
/3. IF EQUAL, CHECKS CURRENT SUBTEST LOOP.

```

5600	COMP,	PAGE	
0000		0	
5601		DCA	BAD
5602		TAD	GOOD
5603		CIA	
5604		TAD	BAD
5605		SNA CLA	
5606		JMP	+4
5607		TAD	COMP
5610		DCA I	ERRORP
5611		JMP I	ERR1P
5612		TAD	TRACE
5613		SMA CLA	COMP
5614		JMP	
5615		LAS	
5616		AND	K400
5617		SZA CLA	
5620		JMP	COMP
5621		TAD	COMP
5622		DCA I	ERRORP
5623		TAD I	COMP
5624		DCA I	TRPP
5625		TAD	K4324
5626		DCA I	ERMSSGP
5627		TAD	K2252
5630		JMS I	PRNTP
5631	COMP,	ISZ	COMP
5632		LOOP5	
5633		SKP	
5634		ISZ	COMP
5635		JMP I	COMP
5636		ERRORS+1	
5637	ERR1P,	PRNT	
5640	TRPP,	FRAPTR	
5641	K2252,	2252	
5642	K4324,	4324	

```

/SAVE AC IN BAD.
/COMPARE GOOD AND BAD.

/ERROR; SET UP AND GO TO
/ERRORS AND DON'T
/RETURN HERE.
/TRACE REQUESTED?

/YES, TRACE INHIBITED?

/NO, GET ERROR POINTER
/AND PUT IN ERRORS;
/GET STAT1 ADDRESS AND
/PUT IN PRNT.

/FAKE AN ERROR PRINTOUT.
/UPDATE RETURN JUMP TO SUBTEST LOOP JUMP.
/SUBTEST LOOP?

/NO, UPDATE ONE MORE.
/EXIT.

```

/SAVE AND RESTORE COMMON SUBROUTINE INFORMATION ROUTINES,  
 /LOCATIONS TYPE, TYPTM, CRLF, AMGR, AMG8E1, AMG8E2,  
 /AMG8E3, EDIT, EDTEM, EDTEH1, EDTEH2, EDTEH3, ETC. MUST BE SAVED AND RESTORED  
 /SINCE MONITOR MAY INTERRUPT THESE ROUTINES  
 /AND USE THEM ITSELF.

6000	SAVEP,	0200	CLA	
6001		7200	TAD	M16
6002		1242	DCA	SAVEP1
6003		3243	TAD	SRPTS
6004		1266	DCA	SAVEP2
6005		3244	TAD	SRHOLD
6006		1247	DCA	SAVEP3
6007		3245	TAD	SAVEP2
6010		2244	DCA	SAVEP3
6011		2245	ISE	SAVEP2
6012		1644	TAD I	SAVEP4
6013		3246	DCA	SAVEP4
6014		1646	TAD I	SAVEP4
6015		3645	DCA I	SAVEP3
6016		2243	ISE	SAVEP1
6017		5210	JMP	107
6020		5600	JMP I	SAVEP
6021		0000		
6022	RESTP,	7200	CLA	
6023		1242	TAD	M16
6024		3243	DCA	SAVEP1
6025		1247	TAD	SRHOLD
6026		3244	DCA	SAVEP2
6027		1266	TAD	SRPTS
6030		3245	DCA	SAVEP3
6031		2244	ISE	SAVEP2
6032		2245	ISE	SAVEP3
6033		1645	TAD I	SAVEP3
6034		3246	DCA	SAVEP4
6035		1644	TAD I	SAVEP2
6036		3646	DCA I	SAVEP4
6037		2243	ISE	SAVEP1
6040		5231	JMP	107
6041		5621	JMP I	RESTP
6042		7762		
6043	M16,	0000	-16	
6044	SAVEP1,	0000		
6045	SAVEP2,	0000		
6046	SAVEP3,	0000		
6047	SAVEP4,	0000		
6047	SRHOLD,	0047		
6050		0000		
6051		0000		
6052		0000		
6053		0000		
6054		0000		
6055		0000		
6056		0000		
6057		0000		

6060	3000	0
6061	3000	0
6062	3000	0
6063	3000	0
6064	3000	0
6065	3000	0
6066	6066	0
6067	5270	0
6070	6244	0
6071	6261	0
6072	6200	0
6073	6237	0
6074	6214	0
6075	6240	0
6076	6301	0
6077	6326	0
6100	6327	0
6101	6330	0
6102	6331	0
6103	6547	0
6104	6561	0

SRPTS:

CRLE  
TYPE  
TYPIEM  
AMG8E  
AMG8E1  
AMG8E2  
AMG8E3  
EDIT  
EDTEM  
EDTEM1  
EDTEM2  
EDTEM3  
SRBSW  
SRBSWT

/LOOP SWITCH SENSORS.  
LAS5,    Ø    LAS  
          AND   K127  
          SNA CLA  
          ISZ   LAS5  
          JMP I   LAS5  
LAS6,    Ø    LAS  
          AND   K40  
          SNA CLA  
          ISZ   LAS6  
          JMP I   LAS6  
LAS7,    Ø    LAS  
          AND   K20  
          SNA CLA  
          ISZ   LAS7  
          JMP I   LAS7  
LAS8,    Ø    LAS  
          AND   K10  
          SNA CLA  
          ISZ   LAS8  
          JMP I   LAS8  
LAS9,    Ø    LAS  
          AND   K4  
          SNA CLA  
          ISZ   LAS9  
          JMP I   LAS9  
K4,       4

6105 0000  
6106 7604  
6107 0027  
6110 7650  
6111 2305  
6112 5705  
6113 0000  
6114 7604  
6115 0025  
6116 7650  
6117 2313  
6120 5713  
6121 0000  
6122 7604  
6123 0024  
6124 7650  
6125 2321  
6126 5721  
6127 0000  
6130 7604  
6131 0023  
6132 7650  
6133 2327  
6134 5727  
6135 0000  
6136 7604  
6137 0343  
6140 7650  
6141 2335  
6142 5735  
6143 0004

/"LOOP6"

/"LOOP7"

/"LOOP8"

/"LOOP9"

/PP08-E PACKED ANSCII MESSAGE GENERATOR.  
/ENTERED WITH:  
/JMS AMG8E  
/MSGNP (MESSAGE POINTER)  
/

/00 IS TERMINATOR, 43 (#) IS CARRIAGE RETURN, LINEFEED,  
/THE FOLLOWING CHARACTERS ARE NOT ALLOWED: @ WHICH GIVES  
/TERMINATOR CODE; # WHICH RESULTS IN CRLF; AND 204-207,  
/212, 215, 375, 377, 233,  
/

/EXITS WITH AC CLEAR.  
PAGE

6200	AMG8E, 0	CLA CLL	
6201	7300	TAD I	AMG8E
6202	1600	DCA	AMG8E1
6203	3237	ISZ	AMG8E
6204	2200	TAD I	AMG8E1
6205	1637	BSW	
6206	4522	JMS	AMG8E2
6207	4214	TAD I	AMG8E1
6210	1637	JMS	AMG8E2
6211	4214	ISZ	AMG8E1
6212	2237	JMP	,=6
6213	5205	AMG8E2, 0	AND
6214	0000		K77
6215	0241	SNA	
6216	7450	JMP I	AMG8E
6217	5600	DCA	AMG8E3
6220	3240	TAD	AMG8E3
6221	1240	TAD	M43
6222	1243	SZA CLA	
6223	7640	JMP	,*3
6224	5227	JMS I	CRLFPP
6225	4466	JMP I	AMG8E2
6226	5614	TAD	AMG8E3
6227	1240	TAD	M40
6230	1242	SPA CLA	
6231	7710	TAD	K100
6232	1027	TAD	K200
6233	1030	TAD	AMG8E3
6234	1240	JMS I	TYPEP
6235	4470	JMP I	AMG8E2
6236	5614	AMG8E1, 0	
6237	3000	AMG8E3, 0	
6240	0000	K77, 77	
6241	0077	M40, -40	
6242	7740	M43, -43	
6243	7735		

/TYPE SUBROUTINE,  
/ENTER WITH ANSCII IN AC, EXIT WITH AC CLEAR  
TYPE, 0 DCA TYPTEM /SAVE ANSCII,

6244 0022  
6245 3261

6246 6041 TSF  
 6247 7410 SKP  
 6250 5254 JMP  
 6251 1173 TAD  
 6252 7700 SMA CLA  
 6253 5246 JMP  
 6254 3173 DCA  
 6255 1261 TAD  
 6256 6046 TFS  
 6257 7200 CLA  
 6260 5644 JMP I  
 6261 0000 TYPEM, 0

/TYPE CHARACTER IF REAL TIO FLAG  
 /IS SET OR SOFTWARE FLAG SET,

/RING ITY BELL,

6262 0000 BELL, 0  
 6263 7200 CLA  
 6264 1267 TAD  
 6265 4470 JMS I  
 6266 5662 JMP I  
 6267 0207 K207, 207

/CARRIAGE RETURN-LINE FEED,

6270 0000 CRLF, 0  
 6271 7200 CLA  
 6272 1031 TAD  
 6273 4470 JMS I  
 6274 1300 TAD  
 6275 4470 JMS I  
 6276 4470 JMS I  
 6277 5670 JMP I  
 6300 0212 K212, 212

/EDIT ROUTINE,

/CONVERTS A 4 DIGIT OCTAL NUMBER IN AC TO 2 PACKED ANSCII  
 /WORDS, EXITS WITH LEAST SIGNIFICANT PORTION IN AC  
 /AND MOST SIGNIFICANT IN "EDTEM",

6301 0000 EDIT, 0  
 6302 3327 DCA EDTEM1  
 6303 1327 TAD EDTEM1  
 6304 4522 BSW  
 6305 4312 JMS EDIT1  
 6306 3326 DCA EDTEM  
 6307 1327 TAD EDTEM1  
 6310 4312 JMS EDIT1  
 6311 5701 JMP I EDIT  
 6312 0000 EDIT1, 0  
 6313 3330 DCA EDTEM2  
 6314 1330 TAD EDTEM2  
 6315 0022 AND K7  
 6316 3331 DCA EDTEM3  
 6317 1330 TAD EDTEM2  
 6320 7006 RTL  
 6321 7004 HAL  
 6322 0333 AND K700

6323	1331		TAD	EDTEM3
6324	1332		TAD	K6060
6325	5712		JMP 1	EDIT1
6326	2000		Z	
6327	2000		EDTEM1, 0	
6330	2000		EDTEM2, 0	
6331	3000		EDTEM3, 0	
6332	6060		K6060, 6060	
6333	0700		K700, 700	

/ROUTINES TO CLEAR ALL FLAGS SAFELY.

```

6400 CLR1, PAGE
6401 JMS I INSRSP /"CLEAR1"
6402 JMS I INSRSP /NULL EF AND MTF INTERRUPTS,
6403 JMS I INSRSP
6404 CLR2, CLR2 /DO "CLEAR2"
6405 JMS I CLR1 /"CLEAR2"
6406 JMS I CLR1 /DO "CLEAR3"
6407 CLR3 /SET INTERRUPT ACTIVE FLAG
6408 JMS I CLR1 /INTERRUPT ON,
6409 CLR3, CLR3 /"CLEAR3"
6410 CLA CLL /ENSURE TIO SOFTWARE FLAG
6411 TAD TIOFLG /IS SET PRIOR TO CAF
6412 SZA CLA /GENERATE "INITIALIZE"
6413 JMP I CLR2
6414 CLA CLL
6415 TAD TIOFLG
6416 SZA CLA
6417 JMP I CLR2
6418 TSE
6419 JMP I CLR2
6420 CLA CLL
6421 TAD TIOFLG
6422 SZA CLA
6423 JMP I CLR2
6424 DCA CLL
6425 IOP
6426 LDBR
6427 CLT
6428 DCA TIOFLG
6429 JMP I CLR3
6430 FUNCT
6431 CLR3
6432
6433

```

/SINCE "INITIALIZE WILL NOT CLEAR OB,  
/ZERO FR TEMP,



/FUNCTION SUBROUTINES.

/TO ENABLE RECALL OF BIT 6 IN FR, EACH TIME THE FR IS LOADED  
 /IN THE FUNC ROUTINE, THE FR CONSTANT LOADED IS SAVED IN CORE, IF INITIALIZE  
 /IS GENERATED, THIS CORE LOCATION IS SERVED.

6432	0000	OFFLINE, 0	/OFF LINE, (OFFLIN)
6433	4311	JMS	FUNC
6434	0000	0	
6435	5632	JMP I	OFFLINE
6436	0000	0	/REWIND, (REWIND)
6437	4311	JMS	FUNC
6440	1000	1000	
6441	5636	JMP I	REWIND
6442	0000	0	/READ, (READ)
6443	4311	JMS	FUNC
6444	2000	2000	
6445	5642	JMP I	READ
6446	0000	0	/HEAD COMPARE (RDCOMP)
6447	4311	JMS	FUNC
6450	3000	3000	
6451	5646	JMP I	RDCOMP
6452	0000	0	/WRITE, (WRITE)
6453	4311	JMS	FUNC
6454	4000	4000	
6455	5652	JMP I	WRIT
6456	0000	0	/WRITE EOF, (WEOF)
6457	4311	JMS	FUNC
6460	5000	5000	
6461	5656	JMP I	EOFW
6462	2000	2000	/SPACE FORWARD (SPCFWD)
6463	4311	JMS	FUNC
6464	6000	6000	
6465	5662	JMP I	SPFORD
6466	0000	0	/SPACE REVERSE, (SPCREV)
6467	4311	JMS	FUNC
6470	7000	7000	
6471	5666	JMP I	SPREV
6472	0000	0	/SET EXTENDED GAP BIT IN CONSTANT / (EXTGAP)
6473	7200	7200	
6474	1032	TAD	K400
6475	3277	DCA	KXGAP
6476	5672	JMP I	XGAP
6477	0000	0	
6500	0000	0	/SET GO BIT NOT CHANGING FR /OTHERWISE, (GO)
6501	7200	7200	
6502	1324	TAD	FUNC
6503	2310	AND	K7600A
6504	1027	TAD	K100
6505	4500	LFGR	
6506	7200	7200	
6507	5700	5700	
6510	7600	7600	
			GOE
			K7600A, 7600

6511 2000 /LOAD FR WITH FUNCTION +  
6512 7200 /EXTENDED GAP BIT CONSTANT

6513 1711 CL A  
6514 1277 TAD I  
6515 3324 TAD  
6516 1324 DCA  
6517 4500 TAD  
6520 7200 LFGR  
6521 3277 CL A  
6522 2311 DCA  
6523 5711 ISZ  
6524 0000 JMP I

6513 1711 FUNC  
6514 1277 KXGAP  
6515 3324 FUNC  
6516 1324 FUNC  
6517 4500 /SAVE FUNCTION,  
6520 7200  
6521 3277 KXGAP  
6522 2311 FUNC  
6523 5711 FUNC  
6524 0000

FUNCT, 2

/LOAD THE COMMAND REGISTER WITH THE CONSTANT IN THE CALLING SEQUENCE,  
 /ENTERED BY:  
 /COMAND  
 /"CONSTANT"

EQUIVALENT JMS I CMDP  
 CONSTANT TO BE LOADED

```

6525 0000 0      /"COMAND"
6526 7200  CLA   /GET CONSTANT,
6527 1725  TAD I  CMD  /LOAD,
6530 4477  LCMR
6531 7200  CLA   /ENSURE AC CLEAR,
6532 2325  ISZ   /UPDATE RETURN,
6533 5725  JMP I  CMD  /EXIT.
  
```

/ISSUE ONE BREAK REQUEST  
 /ENTERED BY "BREAK".

```

6534 0000  DATBRK, 0      /"BREAK"
6535 4521  SBRM
6536 7000  NOP
6537 7000  NOP
6540 5734  JMP I  DATBRK
  
```

/GENERATE CONDITIONS NECESSARY  
 /FOR WCOVER, SET GO PIT, WC TO  
 /7777 " LEAVE REMAINDER OF FR ALONE.

```

6541 0000  WCOVER, 0
6542 4550  GO
6543 7240  CLA CMA
6544 4473  LWCR
6545 4535  BREAK
6546 5741  JMP I  WCOVER
  
```

/BYTE SWAP ROUTINE.

```

6547 0000  SRBSW, 0
6550 7106  CLL  RTL
6551 7006  RTL
6552 7006  RTL
6553 3361  DCA
6554 7004  RAL
6555 1361  TAD
6556 2362  AND
6557 1361  TAD
6560 5747  JMP I
6561 0000  SRBSWT, 0
6562 0077  SRBSWN, 77
6552 7006  DCA
6554 7004  RAL
6555 1361  TAD
6556 2362  AND
6557 1361  TAD
6560 5747  JMP I
6561 0000  SRBSWT, 0
6562 0077  SRBSWN, 77
  
```

TST,	TEST01	TEST02	TEST03	TEST04	TEST05	TEST06	TEST07	TEST10	TEST11	TEST12	TEST13	TEST14	TAL,	TEST TALLIES FOR: TPA
6563	4203													/T02
6564	4600													/T03
6565	1000													/T04
6566	1235													/T05
6567	1400													/T06
6570	1600													/T07
6571	2000													/T10
6572	2200													/T11
6573	3000													/T12
6574	3200													/T13
6575	3400													/T14
6576	3600													
6577	0000													
6600	7766													
6601	0000													
6602	7766													
6603	7766													
6604	7766													
6605	0000													
6606	0000													
6607	0000													
6610	0000													
6611	0000													
6612	7766													
6613	4473													
6614	4475													
6615	4477													
6616	4501													
6617	4500													
6620	4474													
6621	4476													
6622	0000													
6623	0000													
6624	0000													
6625	4503													
6626	7777													
6627	4504													
6630	7777													
6631	4506													
6632	7777													
6633	4510													
6634	7777													
6635	4507													
6636	7600													
6637	4503													
6640	7777													
6641	4504													
6642	7777													
6643	4506													
6644	7777													
6645	4510													
6646	7777													
6647	4541													

/EXEC TEST POINTERS,

TEST TALLIES FOR: TPA

/T02  
/T03  
/T04  
/T05  
/T06  
/T07  
/T10  
/T11  
/T12  
/T13  
/T14

TILLS1, TILLS ARE UTILIZED BY  
LCMR THE TEST 01 LOCAL EXECUTIVE  
LDBR /FOR JOT DETERMINATION,  
LFGR  
CWCR  
CCAR

TILLS1, LCMR  
LDBR  
LFGR  
CWCR  
CCAR

TILLS2, LCMR  
LDBR  
LFGR  
CWCR  
CCAR

TILLS3, RCMR  
RCAR  
RCMR  
RDBR  
RFSR  
RWCR  
RCAR  
RCMR  
RDBR  
RFSR  
RWCR

TILLS1, READ  
RCAR  
RCMR  
RDBR  
RFSR  
RWCR

/FUN ON LIST USED BY

/TEST 03 ("T3EX1");

6650	4542	RDCOMP
6651	4543	WRITE
6652	4537	OFFLIN
6653	4540	REWIND
6654	4544	WEOF
6655	4545	SPCFWD
6656	4546	SPCREV
6657	4541	READ
6660	4542	RDCOMP
6661	4543	WRITE
6662	4544	WEOF
6663	5276	SYMPT, I1
6664	5304	I2
6665	5214	GD
6666	5221	BD
6667	5226	OD
6670	5233	WC
6671	5240	CA
6672	5245	CM
6673	5252	FS
6674	5257	MS
6675	5264	DB
6676	5271	AC

/SYMBOL ROUTINE POINTER TABLE;

/ERROR TABLE FOR ERRORS OCCURRING OUTSIDE OF FORMAL TESTS, (CODE 20X)

6677	4001	FE00A,	4001	/ILLEGAL INTERRUPT IN INTSEV, CURRENT CONTENTS OF
6700	7776		7776	/LOCATION 0 IN BAD,
6701	0002	ER00B,	0002	/BITS 4 AND 5 OF CH DID NOT CLEAR AFTER LCMR0,
6702	1020		1020	/DETECTED IN INSR0,
6703	0003	ER00C,	0003	
6704	7601		7601	

/ERROR TABLE FOR ERRORS OCCURRING IN FORMAL TESTS (CODE 01X-14X),

6705	0101	ER01A,	0101	
6706	1400		1400	
6707	0102	ER01B,	0102	
6710	1400		1400	
6711	0103	ER01C,	0103	
6712	1400		1400	
6713	0104	ER01D,	0104	
6714	5400		5400	
6715	0105	ER01E,	0105	
6716	6000		6000	
6717	0106	ER01F,	0106	
6720	7400		7400	
6721	0107	ER01G,	0107	
6722	6000		6000	
6723	0110	ER01H,	0110	
6724	7400		7400	
6725	0111	ER01I,	0111	
6726	7400		7400	
6727	0112	ER01J,	0112	
6730	3400		3400	
6731	0113	ER01K,	0113	
6732	6000		6000	
6733	0114	ER01L,	0114	
6734	7400		7400	
6735	0115	ER01M,	0115	
6736	5400		5400	
6737	0116	ER01N,	0116	
6740	7400		7400	
6741	0117	ER01O,	0117	
6742	7400		7400	
6743	0120	ER01P,	0120	
6744	7400		7400	
6745	2201	ER02A,	2201	
6746	1300		1300	
6747	2202	ER02B,	2202	
6750	1240		1240	
6751	2203	ER02C,	2203	
6752	1220		1220	
6753	2204	ER02D,	2204	
6754	1202		1202	
6755	2205	ER02E,	2205	
6756	1600		1600	
6757	0301	ER03A,	0301	

6760	1110	ER03B,	1110
6761	0302		0302
6762	1252		1252
6763	0303	ER03C,	0303
6764	1652		1652
6765	0304	ER03D,	0304
6766	1652		1652
6767	0305	ER03E,	0305
6770	1252		1252
6771	0306	ER03F,	0306
6772	1110		1110
6773	0307	ER03G,	0307
6774	1050		1050
6775	0310	ER03H,	0310
6776	1110		1110
6777	0311	ER03I,	0311
7000	1050		1050
7001	2401	ER04A,	2401
7002	1310		1310
7003	0402	ER04B,	0402
7004	1414		1414
7005	2403	ER04C,	2403
7006	1270		1270
7007	2404	ER04D,	2404
7010	1270		1270
7011	0405	ER04E,	0405
7012	1070		1070
7013	2501	ER05A,	2501
7014	1252		1252
7015	2502	ER05B,	2502
7016	1652		1652
7017	2601	ER06A,	2601
7020	1272		1272
7021	0701	ER07A,	0701
7022	1400		1400
7023	0702	ER07B,	0702
7024	1400		1400
7025	0703	ER07C,	0703
7026	1400		1400
7027	0704	ER07D,	0704
7030	1400		1400
7031	0705	ER07E,	0705
7032	1400		1400
7033	1001	ER10A,	1001
7034	1000		1000
7035	1002	ER10B,	1002
7036	0174		0174
7037	1003	ER10C,	1003
7040	0174		0174
7041	1004	ER10D,	1004
7042	4174		4174
7043	1005	ER10E,	1005
7044	0174		0174
7045	1006	ER10F,	1006
7046	0174		0174

7047	1007	ER106,	1007
7050	0174		0174
7051	1010	ER10H,	1010
7052	1574		1574
7053	1011	ER10I,	1011
7054	1574		1574
7055	1012	ER10J,	1012
7056	1574		1574
7057	1013	ER10K,	1013
7060	1574		1574
7061	1017	ER10O,	1017
7062	1574		1574
7063	1020	ER10P,	1020
7064	1574		1574
7065	1021	ER10Q,	1021
7066	1574		1574
7067	1022	ER10R,	1022
7070	1574		1574
7071	1023	ER10S,	1023
7072	1576		1576
7073	1024	ER10T,	1024
7074	1576		1576
7075	1025	ER10U,	1025
7076	1574		1574
7077	1026	ER10V,	1026
7100	1574		1574
7101	1027	ER10W,	1027
7102	1574		1574
7103	1030	ER10X,	1030
7104	1574		1574
7105	1032	ER10Z,	1032
7106	1576		1576
7107	1061	ER101,	1061
7110	1574		1574
7111	1062	ER102,	1062
7112	1574		1574
7113	1101	ER11A,	1101
7114	0004		0004
7115	1102	ER11B,	1102
7116	4004		4004
7117	1103	ER11C,	1103
7120	4114		4114
7121	1104	ER11D,	1104
7122	4114		4114
7123	1105	ER11E,	1105
7124	4014		4014
7125	1106	ER11F,	1106
7126	4014		4014
7127	1201	ER12A,	1201
7130	0024		0024
7131	1202	ER12B,	1202
7132	0134		0134
7133	1203	ER12C,	1203
7134	0134		0134
7135	1204	ER12D,	1204



7136 0034  
 7137 1205  
 7140 0034  
 7141 1301  
 7142 0014  
 7143 1302  
 7144 4014  
 7145 1303  
 7146 1040  
 7147 1304  
 7150 1050  
 7151 1305  
 7152 5110  
 7153 3401  
 7154 1656  
 7155 3402  
 7156 1656  
 7157 1403  
 7160 1056  
 7161 1404  
 7162 1404

0034  
 1205  
 0034  
 1301  
 0014  
 1302  
 4014  
 1303  
 1040  
 1304  
 1050  
 1305  
 5110  
 3401  
 1656  
 3402  
 1656  
 1403  
 1056  
 1404  
 1404

/MESSAGES!

MSG1, TEXT

"#TMBE CONTROL TEST PART 1#MAINDEC=08-DHTMA-A#"

7163 4324  
 7164 1570  
 7165 0540  
 7166 0317  
 7167 1624  
 7170 2217  
 7171 1440  
 7172 2405  
 7173 2324  
 7174 4020  
 7175 0122  
 7176 2440  
 7177 6143  
 7200 4315  
 7201 0111  
 7202 1604  
 7203 4503  
 7204 5560  
 7205 7055  
 7206 0410  
 7207 2415  
 7210 0155  
 7211 0143  
 7212 0000

/MSG2, TEXT

4324  
 1570  
 0540  
 0317  
 1624  
 2217  
 1440  
 2405  
 2324  
 4020  
 0122  
 2440  
 6143  
 4315  
 0111  
 1604  
 4503  
 5560  
 7055  
 0410  
 2415  
 0155  
 0143  
 0000

4324  
 0523  
 2440  
 0000  
 4300

/CR,T  
 /E/S  
 /T/SP  
 /% TEST NUMBER FILL IN.  
 /% END MSG

7220	4323	MSG3,	TEXT	"#SELECTED TESTS DONE#"
7221	4514			
7222	2503			
7223	2405			
7224	3440			
7225	2405			
7226	2324			
7227	2340			
7230	2417			
7231	1605			
7232	4300			

7233	4320	/MSG4,	TEXT	"#PASS NNN#"	/CR,P
7234	0123	MSG4,	0123		/A,S
7235	2340		2340		/S,SP
7236	0000	M4P1,	0		/PASS NUMBER
7237	0000	M4P2,	0		/FILL IN,
7240	0000		0000		
7241	7743	MSG6,	TEXT	"?#"	
7242	0000				

7243	4343	MSG7,	TEXT	"#FATAL ERROR#"
7244	3601			
7245	2401			
7246	1440			
7247	0522			
7250	2217			
7251	2252			
7252	0000			

7253	4333	MSG8,	TEXT	"#["
7254	0000			
7255	4323	MSG9,	TEXT	"#SET SR0 FOR QUICK VERIFY#"
7256	0524			
7257	4023			
7260	2260			
7261	4006			
7262	1722			
7263	4021			
7264	2511			
7265	2313			
7266	4026			
7267	0522			
7270	1106			
7271	3143			
7272	0000			

7273	4305	MSG10,	TEXT	"#EXTENDED MEMORY?(0-/) "
7274	3024			
7275	2516			
7276	3405			
7277	3440			

7300 1505  
7301 1517  
7302 2231  
7303 7750  
7304 6055  
7305 6751  
7306 4000

MSG11, TEXT " NNNN"

MSG11, 4040  
M11P1, 0  
M11P2, 0  
M11P3, 0

/SP,SP  
/CONTENTS OF MEMORY  
/FILL IN.  
/END MSG.

/THE FULL ERROR MESSAGE CALLED BY STA12=777 WOULD BE:

/ERMMSG, TEXT "#\*ERNX PC:NNNN 11: NNNN 12: NNNN GDI NNNN BD: NNNN ODI NNNN  
WC: NNNN CA: NNNN CM: NNNN FS: NNNN MS: NNNN DS: NNNN AC: NNNN#"  
/  
ERMMSG, 0  
ERMSP, 0  
ERMSP2, 0  
ERMSP3, 0  
ERMSP4, 0  
ERMSP5, 0  
/SPACING AND SYMBOL FILL IN.  
/ERROR TYPE  
/TEST TYPE  
/SUBTEST LETTER, SPACE  
/SP,P  
/C,I  
/CONTENTS OF PG.  
/DITTO  
/ALL OTHER INFORMATION INSERTED HERE,

7313 0000  
7314 0000  
7315 0000  
7316 0000  
7317 4020  
7320 0372  
7321 0000  
7322 0000

DECIMAL

ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65  
ZBLOCK 65

7323 0000  
7324 0000  
7325 0000  
7326 0000  
7327 0000  
7330 0000  
7331 0000  
7332 0000  
7333 0000  
7334 0000  
7335 0000  
7336 0000  
7337 0000  
7340 0000  
7341 0000  
7342 0000  
7343 0000  
7344 0000  
7345 0000  
7346 0000  
7347 0000  
7350 0000  
7351 0000  
7352 0000  
7353 0000  
7354 0000  
7355 0000  
7356 0000  
7357 0000

7360	0000	ZBLOCK	65
7361	0000	ZBLOCK	65
7362	0000	ZBLOCK	65
7363	0000	ZBLOCK	65
7364	0000	ZBLOCK	65
7365	0000	ZBLOCK	65
7366	0000	ZBLOCK	65
7367	0000	ZBLOCK	65
7370	0000	ZBLOCK	65
7371	0000	ZBLOCK	65
7372	0000	ZBLOCK	65
7373	0000	ZBLOCK	65
7374	0000	ZBLOCK	65
7375	0000	ZBLOCK	65
7376	0000	ZBLOCK	65
7377	0000	ZBLOCK	65
7400	0000	ZBLOCK	65
7401	0000	ZBLOCK	65
7402	0000	ZBLOCK	65
7403	0000	ZBLOCK	65
7404	0000	ZBLOCK	65
7405	0000	ZBLOCK	65
7406	0000	ZBLOCK	65
7407	0000	ZBLOCK	65
7410	0000	ZBLOCK	65
7411	0000	ZBLOCK	65
7412	0000	ZBLOCK	65
7413	0000	ZBLOCK	65
7414	0000	ZBLOCK	65
7415	0000	ZBLOCK	65
7416	0000	ZBLOCK	65
7417	0000	ZBLOCK	65
7420	0000	ZBLOCK	65
7421	0000	ZBLOCK	65
7422	0000	ZBLOCK	65
7423	0000	ZBLOCK	65

/(DONE BY ASSEMBLER,)

OCIAL

7500	4472	*7500	
7501	4473	T10LS1,	
7502	4475	LWCR	16700
7503	4477	LCAR	
7504	4500	LCMR	
7505	4474	LFGR	
7506	4476	CWCR	
7507	4503	CCAR	
7510	4504	RWCR	
7511	4506	RCAR	
7512	4527	RCMR	
7513	4510	REFR	
		RDBR	

7514	4505	RMSR
7515	4502	16710
7516	4511	16720
7517	4515	SKIR
7520	4514	SKTD
7521	4513	SKOB
7522	4520	SOLE
7523	4516	CLF
7524	4512	SKEF

## /INSTRUCTION DEFINITIONS:

## /SPECIAL EXT MEM BASIC INSTRUCTIONS:

6201 /CHANGE DATA FIELD  
 6224 /READ INSTRUCTION FIELD.  
 6244 /RESTORE MEMORY FIELD.

## /TM8E IOT DEFINITIONS:

## /LOAD IOT'S:

4472 16700=JMS I S6700P /IOT 6700, /AC TO WC, 0 TO AC;  
 4473 LWCR=JMS I SLWCRP /0 TO WC;  
 4474 CWCR=JMS I SCWCRP /AC TO CA, 0 TO AC;  
 4475 LCAR=JMS I SLCARP /0 TO CA;  
 4476 CCAR=JMS I SCCARP /AC TO CM, 0 TO AC;  
 4477 LCMR=JMS I SLCMRP /AC0-3, 5 TO FR, 0 TO AC;  
 4500 LFGR=JMS I SLFGRP /AC TO DB, 0 TO AC  
 4501 LDBR=JMS I SLDBRP

## /READ IOT'S:

4502 16710=JMS I S6710P /IOT 6710, /0 TO AC, WC TO AC;  
 4503 RWCR=JMS I SRWCRP /0 TO AC, CA TO AC;  
 4504 RCAR=JMS I SRCARP /0 TO AC, MS TO AC;  
 4505 RMSR=JMS I SRMSRP /0 TO AC, CM TO AC;  
 4506 RCMR=JMS I SRCMRP /0 TO AC, FR TO AC0-4, GO BIT TO AC5, S1 TO AC6-11.  
 4507 RFSR=JMS I SRFSRP /0 TO AC, DB TO AC;  
 4510 RDBR=JMS I SRDBRP

## /CONTROL IOT'S:

4511 16720=JMS I S6720P /IOT 6720, /SKIP IF ERROR FLAG SET,  
 4512 SKEF=JMS I SSKEFP /SKIP IF CONTROL NOT BUSY,  
 4513 SKCB=JMS I SSKCBP /SKIP IF TRANSPORT DONE,  
 4514 SKTD=JMS I SSKTDP /SKIP IF TUR,  
 4515 SKTR=JMS I SSKTRP /0 TO REGISTERS AND FLAGS IF TUR; IF NOT 0 TO MTF, EF, SR  
 4516 CLF=JMS I SCLFP /POWER CLEAR TRANSPORT,  
 4517 CLT=JMS I SCLTP /SET DATA LATE ERRCH;  
 4520 SDLE=JMS I SSDLEP /SET ONE BREAK REQUEST;  
 4521 SBRM=JMS I SSBRMP

## /INSTRUCTION PSEUDO MNEMONICS DEFINITIONS:

4522 BSW=JMS I SRBSWP /SWAP BITS 0-5 WITH BITS 6-11 IN THE AC.  
 4523 CLEAR1=JMS I CLR1P /GENERATE "INITIALIZE" AND LCMR0, TURN INTERRUPT ON,  
 4524 CLEAR2=JMS I CLR2P /GENERATE "INITIALIZE", LEAVE CM ALONE, TURN INTERRUPT ON,  
 4525 CLEAR3=JMS I CLR3P /GENERATE "INITIALIZE" SAFELY WITH RESPECT TO T10, DO NOT TURN  
 /INTERRUPT ON,  
 4526 LOOP5=JMS I LAS5P /CHECK SR5, IF CLEAR, SKIP NEXT INSTRUCTION,  
 4527 LOOP6=JMS I LAS6P /CHECK SR6, IF CLEAR, SKIP NEXT INSTRUCTION,  
 4530 LOOP7=JMS I LAS7P /CHECK SR7, IF CLEAR, SKIP NEXT INSTRUCTION,  
 4531 LOOP8=JMS I LAS8P /CHECK SR8, IF CLEAR, SKIP NEXT INSTRUCTION.

```

CONTROL TEST PART 1 MAINDEC=08-DHTMA-A=L PAL10 .41 16=DEC=72 13:30 PAGE 34-1
4532 LOOPY=JMS I LAS9P /CHECK SRY, IF CLEAR, SKIP NEXT INSTRUCTION,
4533 COMPAR=JMS I COMPP /COMPARE "GOOD" AND "BAD". GO TO "ERRORS" ON FAILURE, CHECK LOOP 5,
4534 ERROR=JMS I ERRORP /GO DIRECTLY TO ERROR ROUTINE, MAY BEUSED FOR FATAL ERRORS ONLY,
4535 BREAK=JMS I DATBRP /ISSUE ONE BREAK REQUEST (SKRM),
4536 COMMAND=JMS I CMDP /LOAD THE CM WITH THE CONSTANT IN THE NEXT LOCATION,
4537 OFFLIN=JMS I OFFLNP /0000+GAP CONSTANT TO FR,
4540 REWIND=JMS I REWNP /1000+GAP CONSTANT FR,
4541 READ=JMS I READP /2000+GAP CONSTANT TO FR,
4542 RDCOMP=JMS I RDCMP /3000+GAP CONSTANT TO FR,
4543 WRITE=JMS I WRITP /4000+GAP CONSTANT TO FR,
4544 WEOF=JMS I WEOF /5000+GAP CONSTANT TO FR,
4545 SPCFWD=JMS I SPCFORP /6000+GAP CONSTANT TO FR,
4546 SPCREV=JMS I SPCREVP /7000+GAP CONSTANT TO FR,
4547 EXTGAP=JMS I XGAPP /SET GAP CONSTANT TO 2400, CONSTANT CLEARED AFTER NEXT FR LOAD,
4550 GO=JMS I GOEP /SET THE "GO" BIT AND LEAVE REST OF FR ALONE,
4553 WCOV=JMS I WCOVEP /GENERATE WORD COUNT OVERFLOW USING FUNCTION ALREADY IN FR,
4551 LXRR=JMS I XIOT1P /IOT SUBROUTINES FOR TEST 01,
4551 CXXR=JMS I XIOT1P
4552 RXXR=JMS I XIOT2P
$$$

```





Table with 10 columns of 11-digit binary strings. Each row corresponds to a line number from 4000 to 7700. The strings are mostly composed of 1s, with occasional 0s appearing in specific columns and rows.

AC	5271	EDIT1	6312	ER07D	7027	ERMP4	7021
ACLOC	5170	EDITP	0263	ER07E	7031	ERMP4P	5152
ACTFLG	2175	EDIEM	6326	ER101	7107	ERMP5	7322
AMG8E	6220	EDIEM1	6327	ER102	7111	ERMP5P	5153
AMG8E1	5237	EDIEM2	6330	ER10A	7033	ERMP6	5363
AMG8E2	6214	EDIEM3	6331	ER10B	7035	ERMP7	5156
AMG8E3	6240	EDIEMP	0064	ER10C	7037	ERMSG	7313
AMG8EP	0061	EOFP	0144	ER10D	7041	ERMSGP	0067
BAD	1155	EOFW	6456	ER10E	7043	ERMSYM	5364
BD	5221	ER00B	6701	ER10F	7045	ERRIP	5636
BELL	6262	ER00C	6703	ER10G	7047	ERROR	4934
BELLP	0065	ER01A	6705	ER10H	7051	ERRORP	0134
BREAK	4335	ER01B	6707	ER10I	7053	ERRORS	5002
BSW	4522	ER01C	6711	ER10J	7055	ERRPTR	5145
CA	5240	ER01D	6713	ER10K	7057	ES	4527
CCAR	4476	ER01E	6715	ER10L	7061	EXEC	4200
CCF	6201	ER01F	6717	ER10M	7063	EXECFP	0071
CLEAR1	4523	ER01G	6721	ER10N	7065	EXECFX	5907
CLEAR2	4524	ER01H	6723	ER10O	7067	EXECL1	4256
CLEAR3	4525	ER01I	6725	ER10P	7071	EXECL2	4245
CLF	4516	ER01J	6727	ER10Q	7073	EXECP	0055
CLR1	6400	ER01K	6731	ER10R	7075	EXITFL	0174
CLRIP	0123	ER01L	6733	ER10S	7077	EXOMT	4303
CLR2	6406	ER01M	6735	ER10T	7101	EXTAL	4354
CLR2P	0124	ER01N	6737	ER10U	7103	EXTEMP	4362
CLR3	6414	ER01O	6741	ER10V	7105	EXTGAP	4547
CLR3P	0125	ER01P	6743	ER10W	7113	FATERR	5031
CLY	4517	ER02A	6745	ER10X	7115	FE00A	6677
CM	5245	ER02B	6747	ER10Y	7117	FS	5252
CMD	5525	ER02C	6751	ER11C	7121	FUNC	6511
CMDB	0136	ER02D	6753	ER11D	7123	FUNCP	0056
COMMAND	4536	ER02E	6755	ER11E	7125	FUNCT	6524
COMP	5600	ER03A	6757	ER11F	7127	GO	5214
COMPAR	4533	ER03B	6761	ER12A	7131	GO	4950
COMPF	5631	ER03C	6763	ER12B	7133	GOE	6500
COMPP	4133	ER03D	6765	ER12C	7135	GOEP	0150
CRLEP	6270	ER03E	6767	ER12D	7137	GOOD	0154
CRLEPP	0066	ER03F	6771	ER12E	7141	HALTC	5211
CS	4420	ER03G	6773	ER13A	7143	HALTCP	5213
CCGR	4474	ER03H	6775	ER13B	7145	I1	5276
CXHR	4551	ER03I	6777	ER13D	7147	I112	5312
CATBRK	6534	ER04A	7001	ER13E	7151	I1P	3145
CATBRP	2135	ER04B	7003	ER14A	7153	I2	5304
CB	5264	ER04C	7005	ER14B	7155	I6700	4472
CBR	4665	ER04D	7007	ER14C	7157	I6710	4502
CLF	4636	ER04E	7011	ER14D	7161	I6720	4511
CLF1	4640	ER05A	7013	ERMP1	7314	INSRS	5463
CLF2	4621	ER05B	7015	ERMP1P	5146	INSRSF	5525
CS	4600	ER06A	7017	ERMP2	7315	INSRSP	0060
CS1	4667	ER07A	7021	ERMP2P	5147	INTAGN	5427
CS2	4670	ER07B	7023	ERMP3	7316	INTEF	0164
CS3	6301	ER07C	7025	ERMP3P	5150	INTFLG	2172

INTMTF	0166	KRMF	4347	M42P	4563	RDCMP	6446
INTNGA	5432	KXGAP	6477	MS	0041	RDCMPP	2142
INTOK	5444	LAS5	6105	MARK1	1622	RDCOMP	4542
INTOKP	4544	LAS5P	0126	MARK1P	1762	READ	4541
INTSEV	4345	LAS6	6113	MARK2	1927	READD	6442
INTSEV	5400	LAS6P	0127	MARK2P	1763	READDR	2141
IOT1	2566	LAS7	6121	MARK3	1624	RESTP	6021
IOT1P	5303	LAS7P	0130	MEMFLD	0167	RESTPF	4541
IOT2	0971	LAS8	6127	MONIT	4900	REWIND	4540
IOT2P	5311	LAS8P	0131	MONITP	0062	REWIND	6436
K1	0020	LAS9	6135	MONP	4676	REWNP	2140
K10	0023	LAS9P	0132	MS	5257	REX	4240
K100	0027	LCAR	4475	MSG1	7163	REXP	4536
K1000	5366	LDBR	4477	MSG10	7273	RFSR	4507
K17	5165	LDBR	4501	MSG11	7307	RIF	6224
K2	0021	LFGR	4500	MSG2	7213	RMF	6244
K20	0024	LISN	4677	MSG3	7220	RMSR	4505
K200	0030	LISN1	4712	MSG4	7233	RWCR	4503
K2000	0033	LISN2	4724	MSG6	7241	RXXR	4552
K207	6267	LISN3	4740	MSG7	7243	S670W	4000
K212	6300	LISNP	4543	MSG8	7253	S670PP	0072
K215	0031	LISNT1	4746	MSG9	7255	S6710	4040
K2252	5641	LISNUM	4727	NOEXLP	4336	S6710P	0102
K3	0005	LOOP5	4526	NOTSPH	4277	S6720P	0111
K377	5370	LOOP6	4527	NOTSRN	4315	SAVEAC	0162
K3774	2051	LOOP7	4530	OD	5226	SAVEL	0163
K4	6143	LOOP8	4531	OFFLIN	4537	SAVEP	6000
K40	0025	LOOP9	4532	OFFLINE	6432	SAVEP1	6043
K400	0032	LWCR	4473	OFFLNP	0137	SAVEP2	6044
K4000	0034	LXXR	4551	OLD	0156	SAVEP3	6045
K4040	0035	M1	5461	P0	5462	SAVEP4	6046
K4300	5166	M11P1	7310	P1	4351	SAVEPP	4540
K4324	5642	M11P1P	4671	P2	4352	SBRM	4521
K4343	5167	M11P2	7311	P3	4353	SCCAR	4020
K4352	5170	M11P2P	4672	PASCNT	4346	SCCARP	0076
K522	5171	M14	0042	PRCNT1	5154	SCDF	4144
K605	5172	M142	5460	PRCNT2	5155	SCDF1	4145
K6060	6332	M16	6042	PRGFLO	0165	SCDFIP	4350
K7	0022	M2	0037	PRMPT	5160	SCDFP	0057
K70	0026	M20	0043	PRNT	5053	SCLF	4124
K700	6333	M233	5457	PRNTCK	5135	SCLFP	0116
K72	5367	M240	4747	PRNTK	5151	SCLT	4130
K7400	0036	M260	0044	PRNTLP	5117	SCLTP	0117
K7600	0574	M270	0045	PRNTP	5637	SCWCR	4010
K7600A	6510	M2P1	7216	QHALT	5200	SCWCRP	0074
K77	6241	M2P1P	4360	QHALTP	5164	SDLE	4520
K7700	5173	M4	0040	GPRNT	5040	SETEF	3157
KCDF	0046	M40	6242	QUES	4673	SETEFP	3345
KJMP61	1757	M43	6243	QUESP	4542	S10T1	0565
KJMP62	1762	M4P1	7236	RCAR	4504	S10T1P	0151
KJMP13	4344	M4P1P	4364	RCMR	4576	S10T2	7570
KNOP	1756	M4P2	7237	ROBR	4512		

SI012P	0152	START	0200	T04T1	1376	T11D	3045
SKCR	4513	SYMADR	5151	T04T2	1377	T11D1	3055
SKEF	4512	SYMBOL	9157	T05A	1402	T11E	3066
SKIPR	4147	SYMPT	6663	T05ASL	1406	T11E1	3074
SKTD	4514	SYMPTP	9162	T05B	1424	T11EX1	3122
SKTR	4515	SYMSEV	9330	T05BSL	1427	T11EX2	3141
SLCAR	4014	SYMTEM	5365	T05LDR	1451	T11EX3	3134
SLCARP	0075	T01A	0211	T05LDS	1450	T11F	3122
SLCMR	4024	T01B	0216	T06A	1602	T11F1	3110
SLCMRP	0077	T01C	0223	T06ADF	1615	T11LS1	3146
SLDBRP	4034	T01D	0235	T06AMP	2002	T12A	3207
SLDBRP	0101	T01E	0243	T07A	2011	T12A1	3215
SLFGR	4030	T01F	0255	T07B	2011	T12B	3225
SLFGRP	0100	T01G	0270	T07C	2021	T12C	3240
SLWCR	4004	T01H	0304	T07D	2032	T12D	3256
SLWCRP	0073	T01I	0313	T07E	2037	T12E	3267
SPCFND	4545	T01J	0323	T101	2637	T12EX1	3305
SPCREV	4546	T01K	0336	T102	2660	T12EX2	3322
SPFORD	6462	T01L	0347	T10A	2204	T12EX3	3317
SPFORP	0145	T01M	0355	T10B	2216	T12EX4	3326
SPREV	6466	T01N	0362	T10C	2231	T12INT	3335
SPREVP	0146	T01O	0403	T10D	2244	T12LS1	3337
SRBSW	6547	T01OL	0400	T10E	2264	T13A	3402
SRBSWN	6562	T01P	0427	T10F	2273	T13B	3415
SRBSWP	0122	T02A	0602	T10G	2302	T13B1	3424
SRBSWT	6561	T02B	0617	T10H	2313	T13C	3434
SRCAR	4050	T02C	0634	T10I	2324	T13D	3447
SRCARP	0104	T02D	0654	T10J	2336	T13E	3461
SRCMR	4060	T02E	0670	T10K	2350	T13E1	3471
SRCMRP	0106	T02LDM	0745	T10KP	2552	T13EX1	3503
SRDBR	4070	T02LDR	0731	T10LPB	2546	T13EX2	3504
SRDBRP	0110	T02LDS	0707	T10LPH	2705	T14A	3602
SRFSR	4064	T02LDT	0716	T10LS1	7500	T14B	3612
SRFSRP	0107	T02LDU	0714	T100	2400	T14BIT	3777
SRHOLD	6047	T02LP9	0704	T10P	2410	T14C	3623
SRMSR	4054	T03A	1004	T10Q	2421	T14C1	3637
SRMSRP	1105	T03B	1015	T10R	2430	T14D	3650
SRPTS	6066	T03C	1027	T10S	2443	T14EX1	3671
SRWCR	4044	T03D	1042	T10T	2454	T14EX2	3677
SRWCRP	0103	T03E	1062	T10U	2500	T14EX3	3701
SSBRM	4140	T03F	1101	T10V	2511	T14EX4	3710
SSBRMP	1121	T03G	1120	T10W	2531	T14EX5	3725
SSDLE	4134	T03H	1141	T10X	2602	T14EX6	3727
SSDLEP	0120	T03I	1156	T10X1P	2361	T14EX7	3747
SSKCB	4105	T04A	1237	T10X2P	2362	T14EX8	3760
SSKCBP	2113	T04AL	1243	T10X3P	2363	T14TM1	3775
SSKEF	4100	T04ASL	1252	T10Z	2613	T14TM2	3776
SSKEFP	1112	T04B	1264	T11A	3003	T1EX1	0450
SSKTD	4112	T04CD	1276	T11B	3011	T1EX1A	0461
SSKTD P	0114	T04CL	1302	T11B1	3016	T1EX1P	2375
SSKTR	4117	T04CSL	1324	T11C	3027	T1EX2	2473
SSKTRP	2115	T04E	1357	T11C1	3037	T1EX2P	0376

T1EX3	0501	TEST04	1235
T1EX3P	0377	TEST05	1400
T1EX4	0205	TEST06	1600
T1EX5	0340	TEST07	2000
T1EX6	0524	TEST10	2200
T1EX7	0516	TEST11	3000
T1EX8	0530	TEST12	3200
T1EX11	0561	TEST13	3400
T1EX12	0562	TEST14	3600
T1EX13	0563	TEST1P	4366
T1EX14	0564	TR	4501
T1LP7	0371	TRACE	0171
T1LP7P	0573	TRPP	5640
T1LS1	6613	TS	4435
T1LS1P	0556	TSI	6563
T1LS2	6620	TSI10P	2704
T1LS2P	0557	TS1P	0447
T1LS3	6625	TSTAEX	4357
T1LS3P	0560	TSTAT	0176
T3EX1	1213	TSTATM	4537
T3EX1P	1176	TSTEM1	4535
T3EX2	1231	TSTNUM	0177
T3EX2P	1175	TSTP	4365
T3EX3	1225	TSTPP	4355
T3EX32	1177	TSTAL	4361
T3LOAD	1200	TTOFLG	0173
T3LODP	2004	TXXTM1	0157
T3LS1	6647	TXXTM2	0160
T4LP6	1262	TXXTM3	0161
T4LP7	1373	TYPE	6244
T5LP6	1422	TYPEP	0070
T5LP7	1445	TYPIEM	6261
T6CNT1	1764	WC	5233
T6CNT2	1765	WCOV	4553
T6CNT3	1766	WCOVER	0153
T6EX1	1642	WCOVER	6541
T6EX2	1664	WEOF	4544
T6EX3	1727	WHAT	5163
T6EX4	1747	WRIT	6452
T6EX5	1737	WRITE	4543
T6EX6	1753	WRITP	0143
T6FLAG	1767	XBUFF	0050
T6I1	1713	XBUFFP	0047
T6SAVE	1761	XGAP	6472
TA	4517	XGAPP	0147
TAL	6577		
TALP	4367		
TALPP	4356		
TCR	4513		
TEST01	0203		
TEST02	0600		
TEST03	1000		

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 35 SECONDS

4K CORE USED